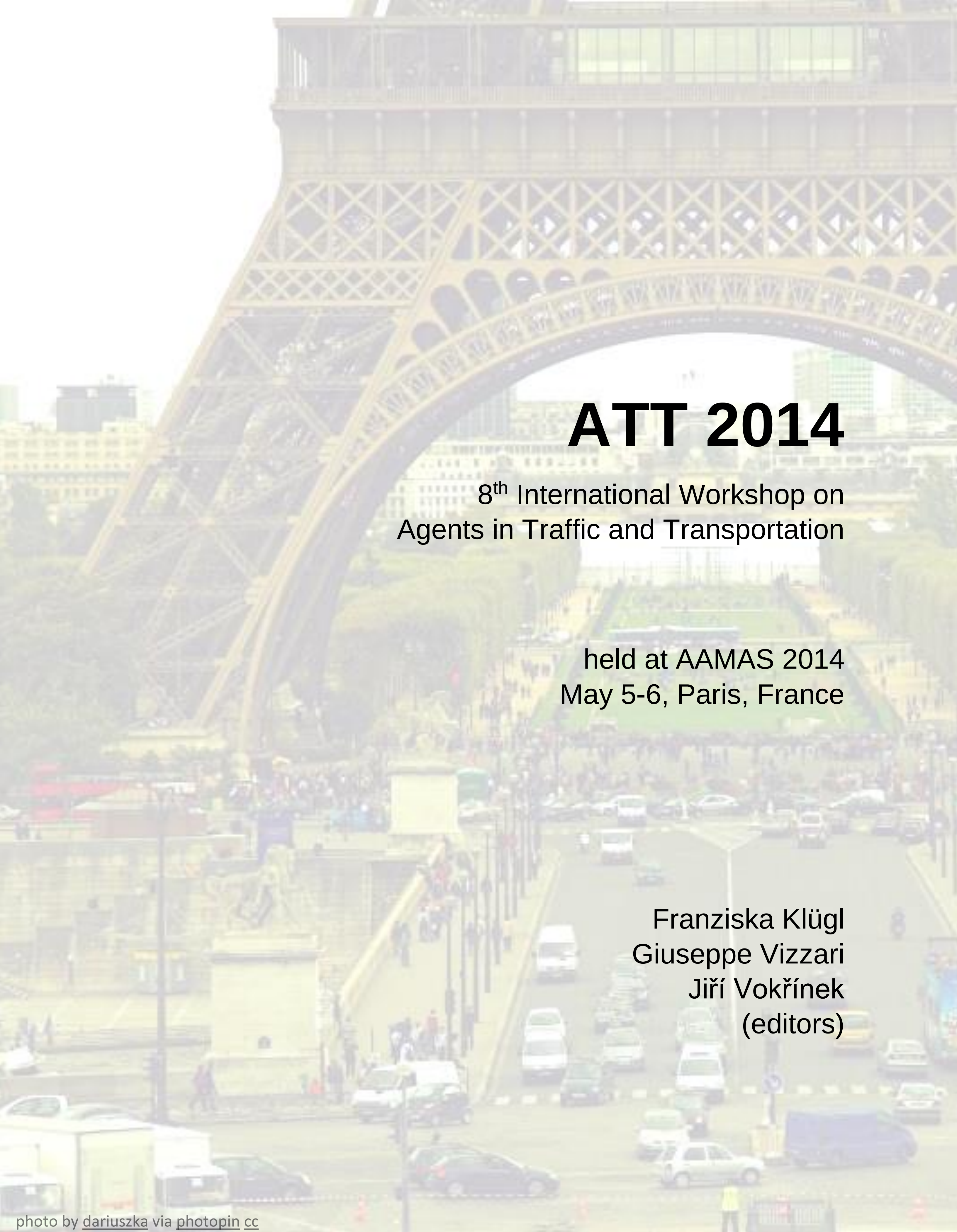


# ATT 2014

8th International Workshop on
Agents in Traffic and Transportation

held at AAMAS 2014
May 5-6, Paris, France

Franziska Klügl
Giuseppe Vizzari
Jiří Vokřínek
(editors)

# Preface

Traffic and transportation became one of the most vivid application areas for multiagent and agent technology. Traffic and transportation systems are not only spatially distributed, but also made up by subsystems with a high degree of autonomy. Consequently, many applications in this domain can be adequately modelled as autonomous agents and multiagent systems.

This is the eight of a well established series of workshops since 2000. The international workshop series on Agents in Traffic and Transportation (ATT) provides a forum for discussion for researchers and practitioners from the fields of artificial intelligence, multiagent systems and transportation engineering. The series aims at bringing researchers and practitioners together in order to set up visions on how agent technology can be used to model, simulate, and manage large-scale complex transportation systems, both at micro and at macro level.

The eight edition of ATT was held together with the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), in Paris (France) on May 5–6. Previous editions were: Barcelona, together with Autonomous Agents in 2000; Sydney, together with ITS 2001; New York, together with AAMAS 2004; Hakodate, together with AAMAS 2006; Estoril, together with AAMAS 2008; Toronto, together with AAMAS 2010; Valencia, together with AAMAS 2012.

This edition of the workshop attracted the submission of 22 high-quality papers. All papers were thoroughly reviewed by at least three renowned experts in the field. Based on the reviewers reports, and the unavoidable space and time constraints associated with the workshop, it was possible to select only 11 of these submissions as full papers and 5 as short papers, leading to an acceptance rate of 50% for full papers. In the process, a number of good and interesting papers had to be rejected.

The present workshop proceedings cover a broad range of topics related to Agents in Traffic and Transportation, tackling the use of tools and techniques based on agent-based simulation, optimization and resource sharing, smart city perspectives, negotiation strategies and pedestrian dynamics. The papers were organized in 5 sessions in two wonderful days. Session 3 of the firs day was dedicated to the inspiring tutorials given by Neila Bhouri (Traffic and Traffic Control – Principles and Engineering) and Jean-Michel Auberlet (Human factors in modeling a simulation for traffic and transportation: examples in pedestrian and driver modeling).

Finally, we owe a big Thank you to all people - authors, reviewers, invited speakers and chairs of the AAMAS conference – who dedicated their time and energy to make this edition of ATT a success.

Paris, May, 2014 Franziska Klügl, Giuseppe Vizzari, Jiří Vokřínek

# Program Committee

Ana L. C. Bazzan (Univ. Federal do Rio Grande do Sul)
Itzhak Benenson (Tel Aviv University)
Ladislau Boloni (University of Central Florida)
Daniel Borrajo (Universidad Carlos III de Madrid)
Eduardo Camponogara (Federal Univ. of Santa Catarina)
Shih-Fen Cheng (Singapore Management University)
Winnie Daamen (Delft University of Technology)
Paul Davidsson (Malmö University)
Bart De Schutter (Delft University of Technology)
Alexis Drogoul (IRD)
Hideki Fujii (The University of Tokyo)
Hiromitsu Hattori (Kyoto University)
Tom Holvoet (K.U. Leuven)
Tomas Klos (Delft University of Technology)
Tobias Kretz (PTV Group)
Rene Mandiau (Université de Valenciennes)
Jörg P. Müller (TU Clausthal)
Yuu Nakajima (Toho University)
Sascha Ossowski (University Rey Juan Carlos)
Omer Rana (Cardiff University)
Nicole Ronald (Swinburne University of Technology)
Rosaldo Rossetti (University of Porto)
René Schumann (UASWS, Switzerland)
Armin Seyfried (Reserch Centre Jülich)
Sabine Timpf (University of Augsburg)
Ronald Van Katwijk (TNO)
Lszl Zsolt Varga (MTA SZTAKI)
Matteo Vasirani (EPFL)
Li Weigang (University of Brasilia)
Tomohisa Yamashita (AIST)

# Organizers

Franziska Klügl (Örebro University)
Giuseppe Vizzari (University of Milano-Bicocca)
Jiří Vokřínek (Czech Technical University in Prague)

# Contents

# An Agent Based Model for the Simulation of Road Traffic and Transport Demand in A Sydney Metropolitan Area

Nam Huynh[*1], Vu Lam Cao[2], Rohan Wickramasuriya[3], Matthew Berryman[4], Pascal Perez[5], Johan Barthelemy[6]

SMART Infrastructure Facility, University of Wollongong, NSW 2522, Australia

[*1]nhuynh@uow.edu.au (corresponding author), Tel: +61 2 4239 2329

[2]vlcao@uow.edu.au, Tel: +61 2 4239 2353

[3]rohan@uow.edu.au, Tel: +61 2 4239 2344

[4]mberryma@uow.edu.au, Tel: +61 2 4221 3303

[5]pascal@uow.edu.au, Tel: +61 2 4252 8238

[6]johan@uow.edu.au, Tel: +61 2 4239 2329

## ABSTRACT
Agent based modelling has emerged as a promising tool to provide planners with sophisticated insights on social behaviour and the interdependencies characterising urban system, particularly with respect to traffic and transport planning. This paper presents an agent based model for the simulation of road traffic and transport demand of an urban area in south east Sydney, Australia. In this model, each agent represents an individual in the population of the study area. Each individual in the model has a travel diary which comprises a sequence of trips the person makes in a representative day as well as trip attributes such as travel mode, trip purpose, and departure time. Individuals in the model are associated with each other by their household relationship, which helps define the interdependencies of their travel diary and constrains their mode choice. This feature allows the model to not only realistically reproduce how the current population uses existing transport infrastructure but more importantly provide comprehensive insight into future transport demands of an urban area. The router of the traffic micro-simulation package TRANSIMS is incorporated in the agent based model to inform the actual travel time of each trip (which agents use in considering new travel modes) and changes of traffic density on the road network. Simulation results show very good agreement with survey data in terms of the distribution of trips done by the population by transport modes and by trip purposes, as well as the traffic density along the main road in the study area.

## Keywords
Agent based model, TRANSIMS, road traffic, transport demand, urban planning

## 1. INTRODUCTION
The ability to realistically predict the demand of transport and traffic on the road network is of critical importance to efficient urban transport planning. Agent based models of urban planning have been increasingly introduced over the last decades. Miller et al. [9] developed model ILUTE (Integrated Land Use, Transportation, Environment) to simulate the evolution of the whole Toronto region in Canada with approximately 2 million households and 5 million people over an extended period of time. Besides giving useful information to analyse a wide range of transport and other urban policies, ILUTE also explicitly models travel demand as an outcome of the integration between individual and household decisions based on activities that they commence during a day. Raney et al. [13] presented a multi-agent traffic simulation for all of Switzerland with a population of around 7 million people. Balmer et al. [1] demonstrated the flexibility of agent based modelling by successfully developing an agent based model that satisfactorily simulate the traffic demands of two scenarios: (i) Zurich city in Switzerland with 170 municipalities and 12 districts and (ii) Brandenburg city in Germany with 1008 traffic analysis zones. Many other agent based models for transport and urban planning can be found in the literature with different geographical scales and at various levels of complexity of agent's behaviours and autonomy [2, 4, 5, 8, 14-19]. They proved that with a large real world scenario, agent based modelling, while being able to reproduce the complexity of an urban area and predict emergent behaviours in the area, has no issue with the performance [17]. They also show that for traffic and transport simulation purposes, agent based modelling has been considered as a reliable and well worth developing tool that planners can employ to build and evaluate alternative scenarios of an urban area.

Many models that have been reported in the literature however are unable to explicitly simulate the dynamic interactions between the population growth, the transport/traffic demands, urban mobility (i.e. residential relocation of households), and the resulting changes in how the population perceive the liveability of an urban area. The agent based model presented in this paper aims at addressing this gap in the literature. The heterogeneity of the population is represented in the model in terms of demographic characteristics, environmental perception, and decision making behaviour. Inherently, the simulated population will evolve over time facilitating the interactions between dynamics of residential relocation of households, transportation behaviours and population growth. Thanks to this feature, the model can be used for exploring long-term (e.g. 20 year time horizon) consequences of various transport and land use planning scenarios.

Individuals are represented in this model as autonomous decision makers that make decisions that affect their environment (i.e. travel mode choice and relocation choice) as well as are required to make decisions in reaction to changes in their environment (e.g. family situation, employment). With respect to transportation, each individual has a travel diary which comprises a sequence of trips the person makes in a representative day as well as trip attributes such as travel mode, trip purpose, and departure time. Individuals in the model are associated with each other by their household relationship, which helps define the interdependencies of their travel diary and constrains their mode choice. This feature, together with the interactions between urban mobility, transportation behaviours, and population growth, allows the model to not only realistically reproduce how the current population uses existing transport infrastructure but more importantly provide comprehensive insight into its future transport demands. The router of the traffic micro-simulation package TRANSIMS is incorporated in the agent based model to inform the actual travel time of each trip (which agents use in considering new travel modes) and changes of traffic density on the road network.

Major components that constitute the agent based model in this study are (i) synthetic population, (ii) residential relocation choice, (iii) perceived liveability, (iv) travel diaries, (v) traffic micro-simulation, and (vi) transport mode choice. These components equip the model with unique features that allows it to be used as a comprehensive tool for assisting integrated travel – land use planning. These components are briefly described in Section 2 in order to provide a full picture of the model features and capabilities. The focus of this paper however will be in reporting the simulation results in regards to road traffic and transport demands (Section 3). The paper closes with discussions on further developments of the model.

## 2. MODEL COMPONENTS

This section provides an overview of the six main components that constitute the agent based model in this study. Details on the model architecture and integration of these components are given in [3].

### 2.1 Synthetic Population

The purpose of the synthetic population is to create a valid computational representation of the population in the study area that matches the distribution of individuals and household as per the demographics from census data. The construction of the synthetic population involves the creation of a proto-population calibrated on socio-demographic information provided by the Australian census data (full enumeration). Different to the majority of existing algorithms for constructing a synthetic population, the algorithm used in this study uses only aggregated data of demographic distributions as inputs, i.e. no disaggregated records of individuals or households (e.g. a survey) are required. The resulting synthetic population is made of individuals belonging to specific households and associated with each other by household relationship.

This initial population is evolved according to annual increments during the simulation period. Each individual and household is susceptible to various demographic (e.g. aging, coupling, divorcing, reproducing of individuals) and economic changes controlled by conditional probabilities. The consequent changes in the structure of households as a result of these processes are also captured. Further details of the algorithms for the construction and evolution of the synthetic population used in this study can be found in [6]. An immigrant population may be added to the existent population of the study area at the end of each simulation step.

### 2.2 Residential Location Choice

Household relocation modelling is an integral part of both the residential and transport planning processes as household locations determine demand for community facilities and services, including transport network demands. The approach used to model residential location choice includes two distinct processes: the decision to relocate, and the process of finding a new dwelling. A multinomial logit model was used to represent the process by which households make decision to relocate. The attributes of this model are change in household income, change of household configuration (e.g. having a newborn, divorced couples, newly wed couples), and the tenure of the household. The HILDA data was used to regress the coefficients associated to each of these attributes needed in the binomial logit model. Further details on the development of the model for triggering household relocation can be found in [12].

Once a household is selected for relocation, the second decision determines where the household will relocate and whether they will be renting or buying a dwelling in the target location, if a suitable a dwelling is found. This process of finding a new dwelling is modelled as a constraint satisfaction process, whereby each household will attempt to find a suitable dwelling based on three factors, affordability, availability, and satisfaction.

### 2.3 Perceived liveability

A significant departure of the current model to other existing approaches is the assumption that residential location choice is based not only on availability and affordability principles but also on the perception that individuals have of the quality of their living environment. The perceived liveability component uses a semi-empirical model to estimate individual levels of attraction to and satisfaction with specific locations. The semi-empirical model is a statistical weighted linear model calibrated on a computer assisted telephone interviewing (CATI) survey data collected in the study area. Further details of this semi-empirical model can be found in [10, 11].

### 2.4 Travel Diaries

Each individual in the synthetic population is assigned with a travel diary which comprises a sequence of trips the person makes in a representative day as well as trip attributes such as travel mode, trip purpose, departure time, origin and destination. Because these details of travel behaviours of the population are not completely available in any single source of data (for confidentiality reasons), the process of assigning travel diaries to individuals comprises two steps. The first step assigns a trip sequence each individual makes in a representative day using the Household Travel Survey data. Details of each trip in this trip sequence include trip purpose, travel mode, and departure time. The second step assigns locations to the origin and destination of each trip in the trip sequence.

#### *2.4.1 Assigning Trip Sequences To Synthetic Population*

The Household Travel Survey (HTS) data was used to assign trip sequences to individuals in the synthetic population. This data is the largest and most comprehensive source of information on individual patterns for the Sydney Greater Metropolitan Area. The data is collected through face to face interviews with approximately 3000-3500 households each year. Details recorded include information of each trip (e.g. departure time, travel time, travel mode, purpose) as well as socio demographic attributes of the interviewed household.

The assignment of trip sequences to the synthetic population comprises two steps. The first step deterministically searches in HTS data for households that best match the household type, the number of children under 15 years old, and the number of adults of a synthetic population household. This deterministic search gradually relaxes the constraints on exact matching conditions so that the search always returns at least one HTS household. The second step randomly selects a HTS household from the list of households identified in stage 1 and assigns travel diary of individuals in the HTS household to those in the synthetic population household. The random selection follows a uniform distribution.

Further details of the algorithms for the assignment of trip sequences to the synthetic population can be found in [7].

### *2.4.2 Assigning Locations To Trips In Trip Sequences*

Once the trip sequences for all the households in the synthetic population are assigned then the following procedure is carried out to assign activity locations to each trip in a sequence. This procedure had to be followed because the HTS data used for this study did not contain activity locations to ensure the confidentiality of the data and so alternative arrangements needed to be made to ensure that each agent was assigned a location of where to go for a particular activity type either inside or outside the study area. In the case of activity locations outside of the study area, main entry and exit points which acted as the origin/destination of trips coming into or going out of the study area. These main entry/exit points are located near where main entry/exit roads pass the boundary of the study area.

Attributes of activity locations in the study area that are available to this study include the geolocations (i.e. coordinates) and the type of the locations. In order to assign specific coordinates to origin and/or destination of a trip, an activity type must first be determined based on the trip purpose. Based on location type and trip mode, a set of coordinates associated with this location type is assigned to the destination. Details of these two processes are given below.

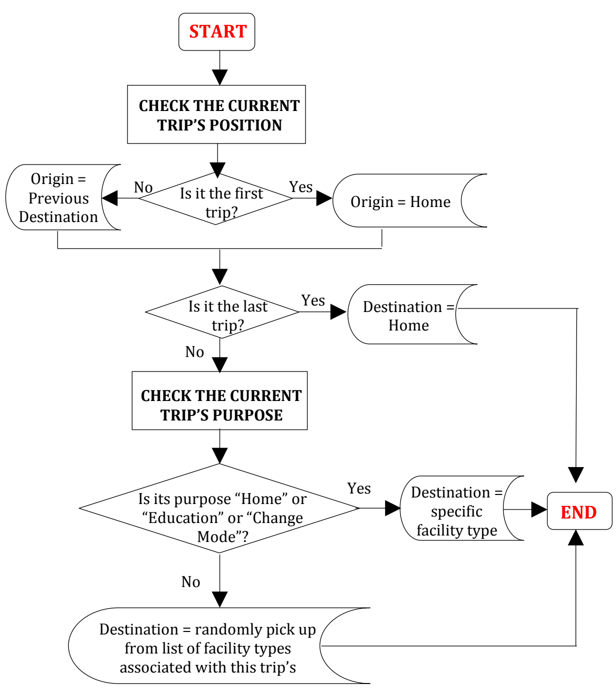



**Figure 1. Flow chart of the assignment of activity types to origin and destination of a trip.**

A flow chart of the assignment of activity types to origin and destination of a trip is shown in Figure 1. The algorithm described in this flow chart applies to all trips of everybody in the population. Depending on the trip purpose, further constraints are applied to correct the assigned activity type. For example, activity types associated with trip purpose "Education" are "Child_care_centre", "Kindergarten", "Education_primary", "Education_school", "Education_university". Choosing which type for the trip destination depends on the age of the individual making that trip.

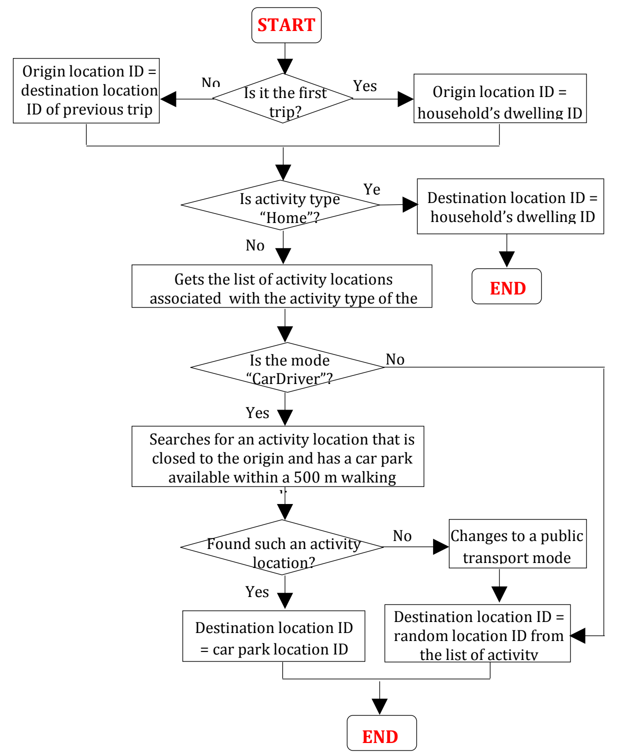



**Figure 2. Flow chart of the assignment of activity locations to origin and destination of a trip.**

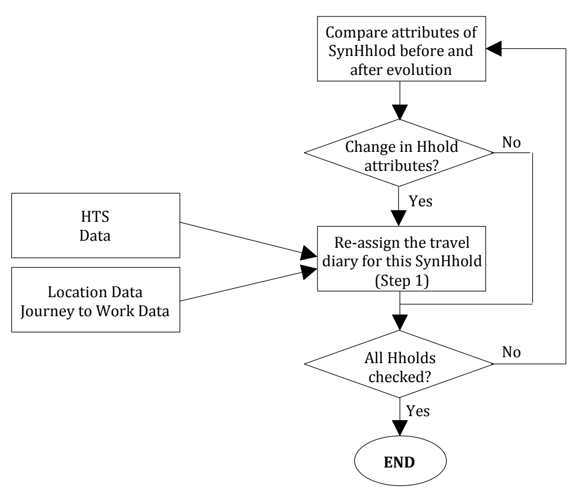



**Figure 3. Travel diaries assignment for successive simulated years.**

flow chart for the assignment of coordinates to trip origin and destination is shown in Figure 2. The algorithm described in this flow chart applies to all trips of everybody in the population. Travel destinations are assigned to account for the constraints of people in the same household travelling together, e.g. destination of a trip of an adult who takes a child to school is similar to the destination of a child. The Journey To Work data is used to assign work locations to work trips. This dataset provides the distribution of trip counts to/from a travel zone from/to another travel zone by each travel mode. For non-work trips (e.g. social and recreational trips), the location of trip destinations is assigned on a random basis.

After each individual has been assigned with a travel diary and specific locations for their trips, corrections to their travel diary may be required to ensure that (i) any children under 15 years old always travel (i.e. have the same modes) with an adult in the household, and (ii) any two individuals who depart and arrive at the same time for the same trip purpose will have the same travel mode and destination. Corrections may also be required to the trip modes of an individual who drives in some trips of his/her travel diary to ensure that a car is used throughout these trips. These corrections are particularly needed after individuals make their travel mode choice (see Section 2.6) during the simulation. This is because the travel mode choice model in itself does not have the visibility of the constraints of co-travelling of individuals in a household nor the connection of trips in an individual's travel diary.

### *2.4.3 Updating Travel Diary Of Individuals During The Simulation*

Sections 2.4.1 and 2.4.2 describe the assigning of initial travel diaries to the synthetic population. Due to changes in the synthetic household attributes (e.g. household type, number of children under 15, etc) as the population evolves, travel diaries may need to be reassigned in subsequent simulation steps to these households in the model. Figure 3 shows the process that is used to reassign/update travel diaries in households whose attributes are different the previous simulation step.

## 2.5 Traffic Micro Simulation

TRANSIMS was chosen as the traffic micro-simulator as, in its current iteration, it is a clean, efficient, C++-based (including good use of STL) platform that supports an individual (person and vehicle) level of modelling, and supports detailed micro-simulation of traffic to support the requirements of our software, including but not limited to:

- road-by-road and minute-by-minute analysis of traffic patterns; and
- details of what individuals are going where on public transport, and analysis of usage (raw, and percentage utilisation).

Normally one would use a process analogous to simulated annealing to arrive at the solution; running the router to establish initial routes, then finding when vehicles jam, and either redirecting them off the street temporarily into a park (if the numbers are sufficiently low) or by then re-routing them using the router and then running the simulation until numbers jammed are sufficiently low. Given the typical travel volumes (around 100,000 commuters), and our desire to simulate a 20-year period, we are forced to run only one typical weekday and weekend in simulation per year, and run only one iteration of the router. We have compared this with test runs of multiple iterations of router and the core micro-simulator of vehicle movements, and found that travel times are within 5%; this we consider sufficient for our purposes.

## 2.6 Transport Mode Choice

The purpose of the travel mode choice algorithm was to accurately describe the decision-making processes of individuals travelling on the transport network in the study area, thus enabling the prediction of the choice of travel modes of individuals in the population. Travel modes considered in this study are car driver, car passenger, public transport, taxi, bicycle, walk, and other.

A multinomial logit (MNL) model was developed for this purpose. At the heart of the MNL formulation is a linear part-worth utility function that calculates the utility of each alternative travel mode choice. Independent variables for this function include the difference of fixed cost and difference of variable cost of the selected travel mode with the cheapest mode. The variable cost is dependent on the estimated travel time, which is the output of the traffic micro-simulation. Another independent variable is the individual's income, acting as a proxy for the individual's perception of value of time. Multinomial logit regression was used on the HTS data to estimate the utility coefficients vector for the possible travel modes.

# 3. SIMULATION RESULTS WITH REGARDS TO TRANSPORT DEMANDS AND ROAD TRAFFIC

The agent based model described in Section 2 is applied to simulate the dynamic interactions between population growth, urban relocation choice and transport demands for Randwick - Green Square, a metropolitan area in south east of Sydney, Australia. This area has a population of approximately 110000 individuals in around 52000 households that live in private dwellings.

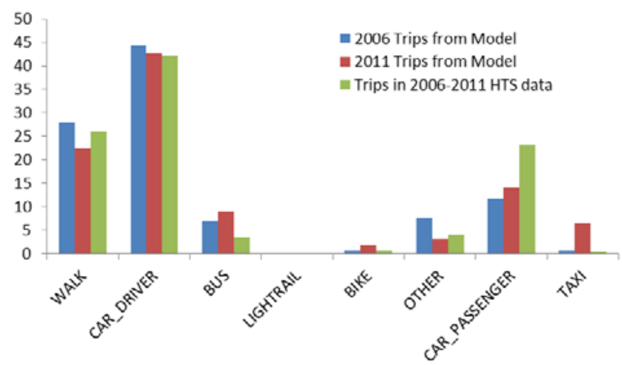



**Figure 4. Percentage of trips by modes from simulation years 2006 and 2011 versus 2006-2011 HTS data.**

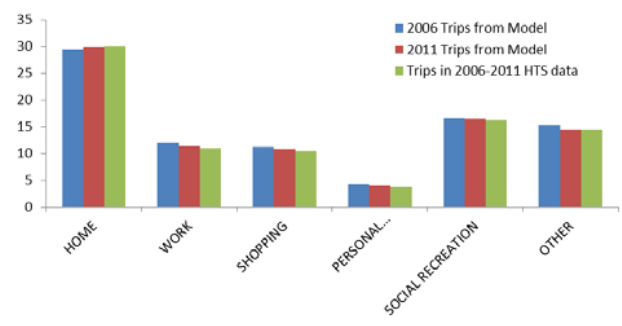



**Figure 5. Percentage of trips by purposes from simulation years 2006 and 2011 versus 2006-2011 HTS data.**

The simulation period is from 2006 to 2011. The initial synthetic population is constructed using the 2006 census data that is available from the Australian Bureau of Statistics. This initial synthetic population was validated that it matches the demographics of the real population at both individual level and household level, and thus is a realistic computational representation of the real population in the area [6]. It was also shown that the synthetic population in year 2011 (i.e. after 5 simulation years) matches the demographics of the population in the study area as described in the 2011 census data. This affirmed that the algorithm to evolve the population while simulating the evolution at individual level can capture the dynamics of household structures in the population.

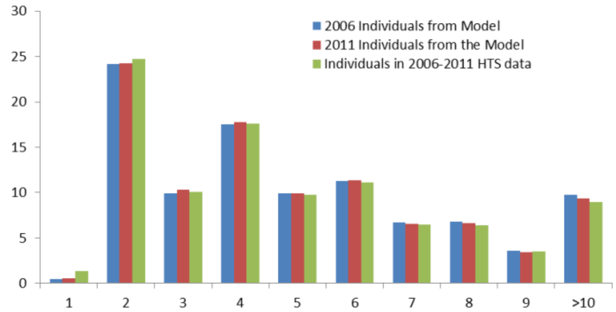



**Figure 6. Percentage of population by number of daily trips for simulation years 2006 and 2011 versus 2006-2011 HTS data.**

Figures 4 and 5 respectively show the percentage of trips by each mode and each purpose with respect to the total number of trips made by the whole population for year 2006 (initial year) and simulation year 2011. Figure 6 compares the percentage of individuals in the synthetic population against that in the HTS data by the number of trips made daily. The distributions in these graphs are in very good agreement with the HTS data for the whole Sydney Greater Metropolitan Area. Please note the HTS data used for comparisons in Figures 4 to 6 is the collective data of years from 2006 to 2011. This is to comply with the suggestion that three or more years of data are pooled to give reliable estimates of travel at a particular geographical level [21].

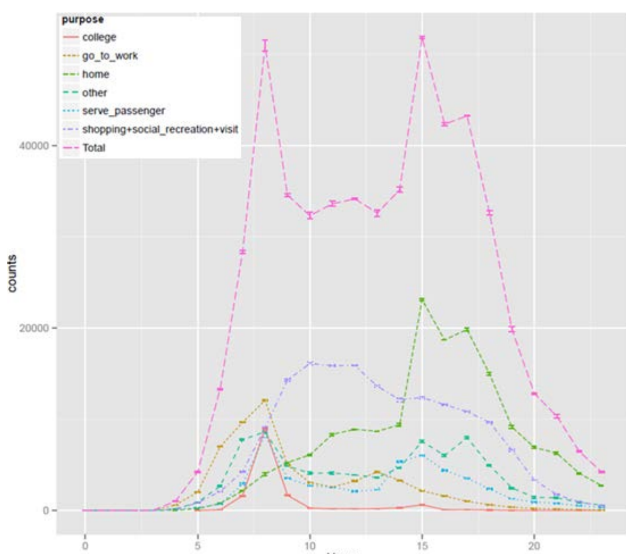
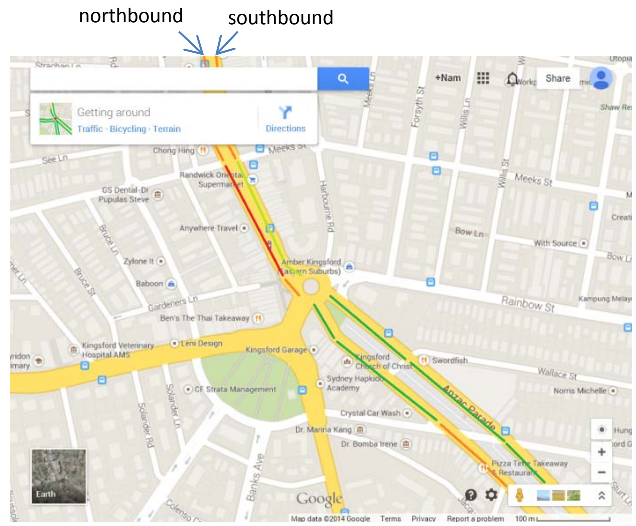
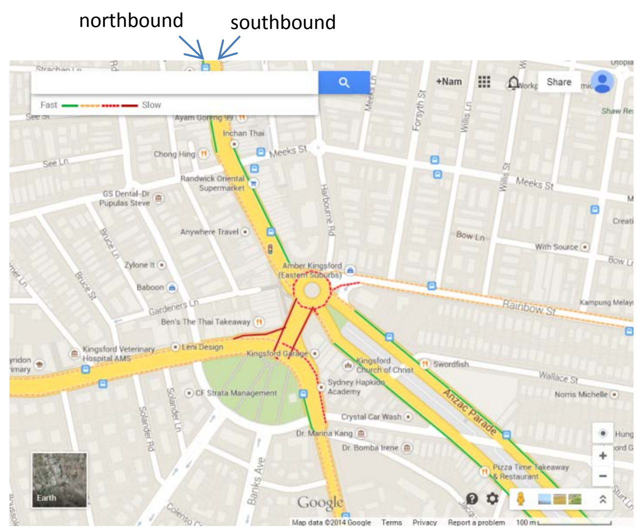



**Figure 7. Trip counts by purposes over 24 hours of a representative day in year 2011.**

Trip counts by purposes over 24 hours of a representative day in year 2011 are shown in Figure 7. In this figure, trips go to work and go to school both peak at 8.00am to 9.00am. Counts of trips go to work however are higher than trips to school at earlier hours (6.00am to 8.00am) which reflects early workers. Trips to work also have a smaller peak between 1.00pm and 2.00pm to reflect trips by people doing afternoon and/or night shifts. Trips for shopping, social activities, recreational and personal services (i.e. 'visit') reach their peak at around 9.00am to 12.00pm and gradually drop in the afternoon. These observations affirm that the model can realistically reproduce the patterns of travel demand of the population in the study area as well as the change of these patterns as the population evolves.




**(a) traffic density from simulation results**

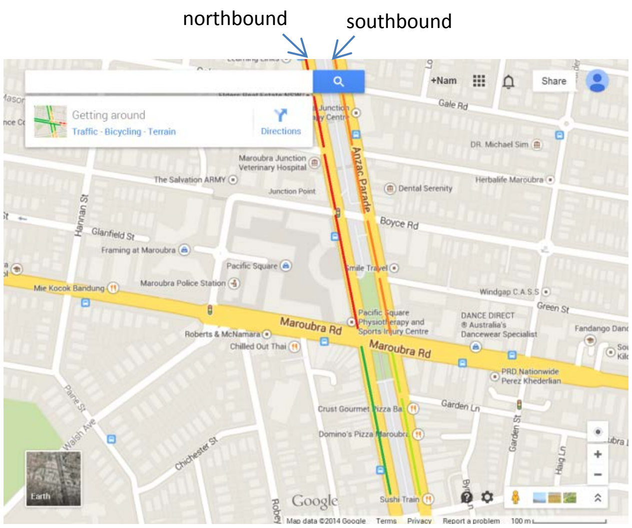



**(b) congestion profile from Google Maps**

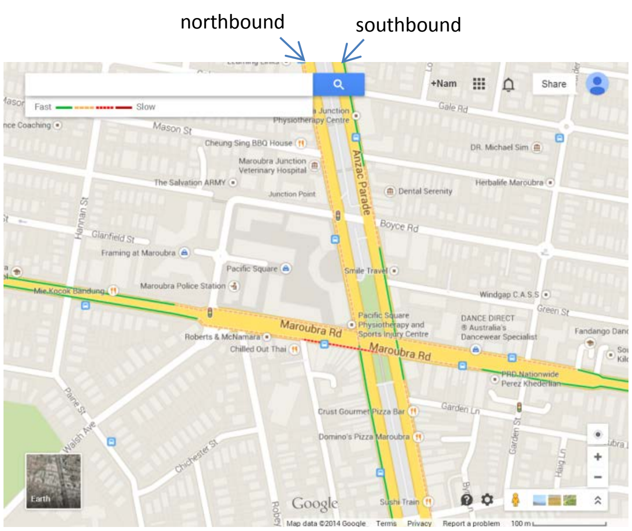
**Figure 8. Traffic density on Anzac Parade near the intersection with Rainbow street in the morning peak hour.**

Traffic density (that was outputted from TRANSIMS router) at two major intersections along Anzac Parade, the main road in the study area, in the morning peak hour (8.00am to 9.00am) compared against their congestion profiles from Google Maps [20] are shown in Figures 8 and 9. The model is able to reproduce the relatively higher northbound traffic density on the part of Anzac Parade north of the intersection with Rainbow street and the part of Anzac Parade north of the intersection with Maroubra Road. The southbound traffic on Anzac Parade at these two locations however is relatively less congested compared to the northbound. These results are in agreement with observations of traffic profiles from Google Maps. Qualitative trends of traffic counts at major cross sections from simulation results were also analysed and in good agreement with survey data in the study area.

Such agreement however does not occur on all parts of the road network. This could be attributed to the randomness in the assignment of activity locations to origin and destination of trips in the travel diaries of the population (see Figure 2). While the assignment of destination locations of trips related to work is constrained by the Journey To Work data, the randomness in assigning destination locations to trips of other purposes does not guarantee a realistic representation of traffic profiles in the model. Note that non-work trips have a significant proportion in the total number of trips made by the population in the study area (see Figures 5 and 7).



**(a) traffic density from simulation results**



**(b) congestion profile from Google Maps**

**Figure 9. Traffic density on Anzac Parade near the intersection with Maroubra Road in the morning peak hour.**

## 4. CONCLUSIONS

This paper has presented an agent based model for the simulation of transport demands and land use for an urban area in south east Sydney, Australia. Being comprised of six major components (synthetic population, residential location choice, perceived liveability, travel diary assignment, traffic micro-simulator, and transport mode choice) the model is able to capture the decision making of the population with respect to relocation and transport, and thus is able to explicitly simulate the dynamic interactions between population growth, transport demands, and urban land use. This is a unique feature that has not been found in many other agent based models for urban transport and urban planning. Thanks to this feature, the model can be used for exploring long-term consequences of various transport and land use planning scenarios.

Various aspects of the simulation results on transport demands of the study area were presented, particularly the percentage of trips by each mode and each purpose with respect to the total number of trips made by the whole population, percentage of population by number of daily trips and the distribution of trips by each purpose over 24 hours of a typical day. Being in good agreement with the corresponding survey data, these results affirm that the model's capability to realistically reproduce travel demand of an urban area and any changes to this travel demand as the population evolves. This is because individuals in the model are associated with each other by their household relationship, which helps define the interdependencies of their travel diary and constrains their mode choice.

Traffic density (from TRANSIMS router) at various locations along the main road in the study area also matches with the observations of traffic congestion on the same road from Google Maps. Mismatches however occur on other (smaller) roads in the study area. This could be attributed to two factors. The first is the lack of a survey data on the origin and destination of non-work trips. The randomness in assigning a location to the destinations of these trips obviously cannot guarantee a realistic representation of traffic demands in the simulation model. The second factor is the limited ability of the TRANSIMS router to realistically reproduce the reasoning of a person in choosing a possible route for the trips the person makes, including dynamic routing to avoid heavy traffic in real time.

## 5. REFERENCES

[1] Balmer, M., Axhausen, K. W., and Nagel, K. 2005. An Agent Based Demand Modeling Framework for Large Scale Micro-Simulation. Working paper, 329, Institute for Transport Planning and Systems (IVT), ETH Zurich, Switzerland.

[2] Benhamza, K., Ellagoune, S., Seridi, H. and Akdag, H. 2010. Agent-based modeling for traffic simulation", International symposium on modeling and implementation of complex systems, *MISC'2010*, pp.219-227.

[3] Berryman, M. J., Denamagage, R. W., Cao, V. & Perez, P. 2013. Modelling and data frameworks for understanding infrastructure systems throuh a systems-of-systems lens. *International Symposium for Next Generation Infrastructure 2013 (ISNGI)*, pp. 1-13.

[4] Cheng, S. F. and Nguyen, T. D. 2011. TaxiSim: A Multiagent Simulation Platform for Evaluating Taxi Fleet Operations. *IAT, IEEE Computer Society*, pp.14-21.

[5] Javanmardi, M., and Mohammadian, A. K. (2012), "Integration of the ADAPTS Activity-Based Model and TRANSIMS", Paper Submitted for Presentation at the 2012 Transport Chicago Conference, 1 June 2012, URL: http://www.transportchicago.org/uploads/5/7/2/0/5720074/ps4_transimsadapts.pdf

[6] Huynh, N., Namazi-Rad, M., Perez, P., Berryman, M. J., Chen, Q. & Barthelemy, J. 2013. Generating a synthetic population in support of agent-based modeling of transportation in Sydney. *20th International Congress on Modelling and Simulation (MODSIM 2013)* pp. 1357-1363.

Australia: The Modelling and Simulation Society of Australia and New Zealand.

[7] Huynh, N. N., Shukla, N., Munoz Aneiros, A., Cao, V. & Perez, P. 2013. A semi-deterministic approach for modelling of urban travel demand. *International Symposium for Next Generation Infrastructure 2013 (ISNGI)*, pp. 1-8.

[8] Maciejewski, M., and Nagel, K. 2013. The influence of multi-agent cooperation on the efficiency of taxi dispatching. VSP Working Paper 13-10. TU Berlin, Transport Systems Planning and Transport Telematics, URL: www.vsp.tu-berlin.de/publications

[9] Miller, E. J., Hunt, J. D., Abraham, J. E., and Salvini, P. A. 2004. Microsimulating urban systems, *Computers, Environment and Urban Systems*, Vol. 28, pp.9–44

[10] Namazi Rad, M., Lamy, F., Perez, P. & Berryman, M. 2012. A heuristic analytical technique for location-based liveability measurement. *The Fifth Annual ASEARC Research Conference: Looking to the future*, pp. 35-38, Wollongong NSW: University of Wollongong.

[11] Namazi Rad, M., Perez, P., Berryman, M. & Lamy, F. 2012. An experimental determination of perceived liveability in Sydney. *ACSPRI Conferences*, *RC33 Eighth International Conference on Social Science Methodology*, pp. 1-13, Australia: ACSPRI.

[12] Namazi Rad, M., Shukla, N., Munoz Aneiros, A., Mokhtarian Dehkordi, P. & Ma, J. 2013. A probabilistic predictive model for residual mobility in Australia. *International Symposium for Next Generation Infrastructure 2013 (ISNGI)*, pp. 1-6, Wollongong: University of Wollongong.

[13] Raney, B., Cetin, N., Vollmy, A., Vrtic, M., Axhausen, K.W. and Nagel, K. 2003. An agent-based microsimulation model of Swiss travel: First results, *Networks and Spatial Economics*, Vol.3, No.1, pp.23– 42.

[14] Scerri, D., Hickmott, S., Bosomworth, K. and Padgham, L. 2012. Using modular simulation and agent based modelling to explore emergency management scenario, *The Australian Journal of Emergency Management*, Vol.27, No.3, pp. 44-48

[15] Schumacher, M., Grangier, L., and Jurca, R. 2008. Modeling and Design of an Agent-based Micro- simulation of the Swiss Highway Network, *Engineering Environment-Mediated Multi-Agent Systems: Lecture Notes in Computer Science*, Vol. 5049, pp.187-203

[16] Shah, A.P., Pritchett, A.R., Feigh, K.M., Kalarev, S.A., Jadlav, A., and Corker, K.M. 2005. Analyzing Air Traffic Management Systems Using Agent-based Modeling and Simulation. *6th USA/Europe ATM R&D Seminar*. 27-30 June 2005, Baltimore, MD, URL: https://smartech.gatech.edu/xmlui/bitstream/handle/ 1853/24616/ ATM2005_Pritchett-etal.pdf

[17] Vogel, A. and Nagel, K. 2005. Multi-agent based simulation of individual traffic in Berlin. *Paper presented at: CUPUM 2005 Conference*, 29 June–1 July, URL: http://128.40.111.250/cupum/searchpapers/papers/ paper179.pdf.

[18] Zhang, B., Chan, W. K., and Ukkusuri, S. V. 2011. Agent-based discrete-event hybrid space modeling approach for transportation evacuation simulation. In *Proceedings of the 2011 Winter Simulation Conference (WSC)*, pp.199-209.

[19] Zhang, L. and Levinson, D. 2004. Agent-Based Approach to Travel Demand Modeling. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1898, TRB, National Research Council, Washington, D.C., pp.28-36.

[20] Website, https://maps.google.com/?ll=47.398349,44.296875&spn=71.56992,183.339844&t=m&layer=t&z=3, accessed on 12/03/2014

[21] Website, http://www.bts.nsw.gov.au/Statistics/Household-Travel-Survey/default.aspx#top, accessed on 31/01/2014

# JADE, TraSMAPI and SUMO: A tool-chain for simulating traffic light control

Tiago M. L. Azevedo Paulo J. M. de Araújo Rosaldo J. F. Rossetti Ana Paula C. Rocha
Artificial Intelligence and Computer Science Lab
Department of Informatics Engineering
Faculty of Engineering, University of Porto, Portugal
{tiago.manuel, paulo.araujo, rossetti, arocha}@fe.up.pt

## ABSTRACT

Increased stress, fuel consumption, air pollution, accidents and delays are some of the consequences of traffic congestion usually incurring in tremendous economic impacts, which society aims to remedy in order to leverage a sustainable development. Recently, unconventional means for modeling and controlling such complex traffic systems relying on multi-agent systems have arisen. This paper contributes to the understanding of such complex and highly dynamic systems by proposing an open-source tool-chain to implement multi-agent-based solutions in traffic and transportation. The proposed approach relies on two very popular tools in both domains, with focus on traffic light control. This tool-chain consists in combining JADE (**J**ava **A**gent **DE**velopment Framework), for the implementation of multi-agent systems, with SUMO (**S**imulation of **U**rban **MO**bility), for the microscopic simulation of traffic interactions. TraSMAPI (**Tra**ffic **S**imulation **M**anager **A**pplication **P**rogramming **I**nterface) is used to combine JADE and SUMO allowing communication between them. A demonstration of the concept is presented to illustrate the main features of this tool-chain, using Q-Learning as the reinforcement learning method for each traffic light agent in a simulated network. Results demonstrate the feasibility of the proposed framework as a practical means to experiment with different agent-based designs of intelligent transportation solutions.

## Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous; I.6 [**Simulation and Modeling**]: Miscellaneous

## General Terms

Algorithms, Design, Experimentation, Verification

## Keywords

MAS, traffic light, JADE, SUMO, TraSMAPI, Q-learning

## 1. INTRODUCTION

Nowadays urban centers face the daily problem of traffic congestion, which in addition to the obvious confusion can create also other negative consequences. Increased stress, fuel consumption, air pollution, accidents and delays are some of these consequences, which society aims to remedy in order to leverage a sustainable development, while mitigating tremendous economic impacts.

Solutions to this problem have evolved over time, more in an immediate response perspective than on a long-term resolution perspective. Initially, the approach was based on the construction of alternative routes with increased capacity. However, available money and territorial area ceased to exist for continuing implementation of this sort of solution. In parallel, traffic lights and roundabouts were introduced but the urban centers continued growth now are demanding more advanced and efficient alternative measures.

The aim of the work described in this paper was to use a tool-chain that allows us to implement a multi-agent system (MAS) for traffic light control. Therefore, a multi-agent system approach was used to answer the daily problem of traffic congestion. This tool-chain consisted in integrating JADE (**J**ava **A**gent **DE**velopment Framework) for controlling the multi-agent system to SUMO (**S**imulation of **U**rban **MO**bility) for traffic simulation. TraSMAPI (**Tra**ffic **S**imulation **M**anager **A**pplication **P**rogramming **I**nterface) was the middleware combining JADE and SUMO and allowing communication between both environments. For the sake of illustration, the implemented agents' learning method was Q-Learning.

As a motivation, just a few simulation tools truly support the concept of agents and multi-agent systems in traffic simulation; MATSim-T [3, 4] and ITSUMO [9, 6] are good examples to be mentioned. However, no standard of wide reach for the implementation of such tools actually exists. Indeed, alternative approaches would require either general purpose MAS-based simulators to be adapted to the specific domain of traffic and transportation, or the other way around with the adaptation of traffic simulators to be adapted so as to support the MAS-based models. With our approach, we expect to benefit from both worlds on an integrated basis. Also, it is important to notice that although SUMO and ITSUMO are both open-source microscopic simulators and have a quite similar acronym, they are no related applications. ITSUMO is a Cellular-Automata-based simulator, whereas SUMO uses a continuous representation of space on road segments. Besides, ITSUMO explicitly consider the metaphor of agents, whereas SUMO can be considered a traditional microscopic simulator, where agents are not explicitly implemented.

The expected contribution of this work, rather than implementing a new agent-based simulator from scratch, adapting or extending existing ones, is to devise an open-source tool-chain to implement MAS-T (MAS in traffic and transportation) on the basis of two very popular tools in both domains. On the one hand, JADE supports the implementation of any MAS solution and, on the other hand, SUMO supports an appropriate representation of the traffic environment in which agents inhabit and perform their tasks.

This paper will start to deeply describe the tools. The conceived model is detailed and instantiated in the proposed tool-chain. An experimental set-up is used to illustrate the proposed approach, followed by the discussion of preliminary results. After discussion on related works, conclusions are drawn as well as are further developments suggested.

# 2. A MAS-BASED TRAFFIC SIMULATION TOOL-CHAIN

The MAS-based traffic simulation tool-chain used consisted in three main tools: JADE, SUMO and TraSMAPI.

A multi-agent system based approach seems to be the appropriate way to represent the different traffic lights in a network. Consequently, it is necessary that a multi-agent system framework take care of the different agent behaviours, as it is the case in JADE.

Next, a microscopic simulator is needed to take care of the traffic road dynamics, such as vehicles decisions. It should be noted that although it is necessary to have vehicles in order to test traffic light control, these vehicles do not need to be modeled as agents. It would be very computationally expensive to simulate a huge quantity of vehicles, each one with driver's decision-making and other cognitive aspects and details. SUMO was the microscopic simulator chosen.

Finally, as traffic lights are considered to be agents, it is necessary they communicate with the simulator. This is important so as to allow their traffic lights in the simulation to have the semaphore plans always updated and agents to perceive the network dynamics. This communication was made through TraSMAPI, consisting of an integration API implemented in Java.

## 2.1 JADE

JADE is a framework completely developed in Java. It simplifies the implementation of multi-agent systems through a middleware that complies with the FIPA[1] specifications and through a set of graphical tools that supports the debugging and deployment phases. The agent platform can be distributed across machines and the configuration can be controlled via a remote GUI [19]. The version used in this work was *4.3.0*, released on March 2013.

One advantage of using JADE to implement MAS is its ability to allow run-time visualisation and control of the interactions among agents in the application. As relevant features for this work, some can be pointed that are not directly connected to agents, that is, are independent of the applications: message transportation, codification and parsing of messages or lifetime of an agent, for instance.

## 2.2 SUMO

SUMO is an open-source program (licenced under GPL[2]) for traffic simulation. Its simulation model is microscopic, that is, each vehicle is explicitly modeled, has its own route and moves individually over the network. It is mainly developed by Institute of Transportation Systems, located at German Aerospace Center [12]. The version used in this work was *0.18.0*, released on August 2013.

Among other features, it allows the existence of different types of vehicles, roads with several lanes, traffic lights, graphical interface to view the network and the entities that are being simulated, and interoperability with other applications at run-time through an API called TraCI. Moreover, the tool is considered to be fast, still allowing a version without a graphical interface where the simulation is accelerated putting aside visual concerns and overheads[12].

In Figure 1 it is possible to visualize the SUMO's graphical interface with a running simulation. It is possible to point out almost all specified features: vehicles stopped at the traffic light as well as a long vehicle entering an intersection.

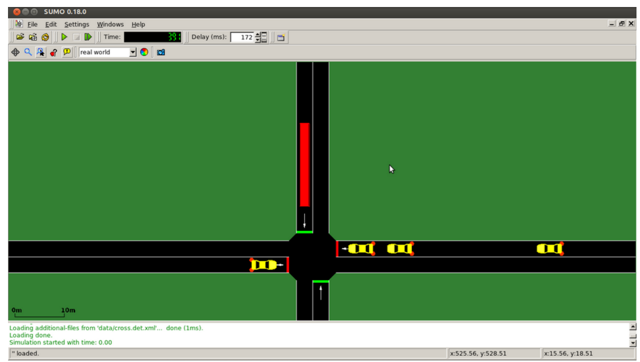



**Figure 1: SUMO working**

This tool was crucial in this work! First, it allows loading different maps (described in XML files) in order to test various scenarios with vehicles and traffic lights. Then, with the simulation itself there is no need to waste time implementing the dynamics of many vehicles and traffic lights, starting soon with the evaluation of algorithms. Finally, interoperability with other applications allows that each agent can be bound to an entity in SUMO, so that changes in the dynamics of traffic lights, for instance, can be visually seen in the SUMO's graphic interface.

## 2.3 TraSMAPI

TraSMAPI can be seen as a generic API for microscopic traffic that allows real-time communication between agents of urban traffic management (such as vehicles and traffic signals) and the environment created by various simulators. This tool was developed in LIACC (Artificial Intelligence and Computer Science Laboratory), University of Porto, having already been tested with two different simulators, including SUMO [20].

This API offers a higher abstraction level than most of microscopic traffic simulators in such a way that the solution is independent from the microscopic simulator to use. Initially, this tool also aimed to gather relevant metrics/statistics and offer an integrated framework for developing multi-agent systems, as shown in Figure 2 [21].

As it can be seen, there were three main modules: a communication module with possibility of various microscopic

[1]**F**oundation of **I**ntelligent **P**hysical **A**gents, an organization that promotes agent-based technology and the interoperability of its standards with other technologies

[2]GNU **G**eneral **P**ublic **L**icense, a free, copyleft license for software and other kinds of works

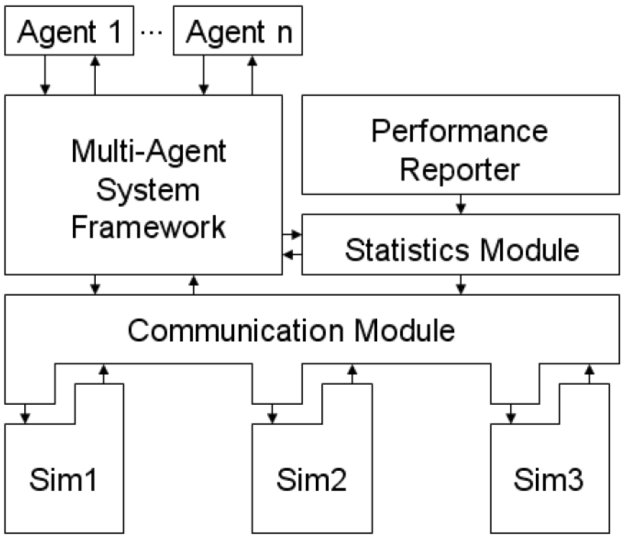



**Figure 2: TraSMAPI's initial architecture**

simulators, the module generating statistics and the module for the MAS management. Presently, only the communication module is functional and this is the module that interests to the scope of the presented work.

## 2.4 The tool-chain

In order to achieve a tool-chain with the previous described tools, it was necessary to extend the TraSMAPI API, enabling to build an abstraction over a SUMO's traffic light entity. Thus, it was necessary to implement the communication protocol regarding the methods of traffic light for value retrieval and state change, in TraCI [23].

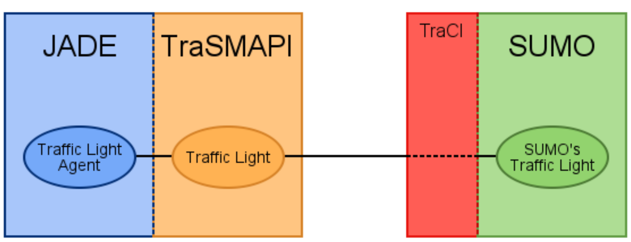



**Figure 3: Communication between JADE and SUMO using TraSMAPI for a traffic light**

The architecture described in Figure 3 shows how it is possible the existence of one or more traffic light agents. Each traffic light agent has a tie to the respective traffic light to be modeled in SUMO. This tie is supported by the TraSMAPI communication module that interacts with the SUMO API, TraCI.

# 3. EXPERIMENTAL SETUP

A simple scenario for the sake of illustration is now described. Although the following scenario is simple and not intended to deeply discuss the appropriateness of implementing traffic control through agents, it illustrates well how our integrated framework could be practically used in this sort of experiments.

## 3.1 Concepts

For the purpose of this work, a traffic light is defined as an intersection that has a semaphore plan, which is characterized by a sequence of phases. Each phase has a duration and a color scheme (green, yellow, flashing yellow and/or red), whose values correspond to every possible maneuver at the intersection. The execution of the phases sequence is called a cycle and has a period equal to the sum of the durations of the phases.

In Figure 4 the intersection has six possible maneuvers, indicated by the arrows, which means that each phase has to specify a color for each maneuver (M1, ..., M6). The sequence of phases is guided by the phase number, and after the end of the sixth phase a 80 cycle duration is completed, following again phase 1. For each maneuver the traffic light may show the green color with symbol **G**, yellow with symbol **y**, flashing yellow with symbol **g** and red with symbol **r**.

| Phase | M1 | M2 | M3 | M4 | M5 | M6 | Duration |
|---|---|---|---|---|---|---|---|
| 1 | G | g | r | r | G | G | 31 |
| 2 | y | g | r | r | y | y | 4 |
| 3 | r | G | r | r | r | r | 6 |
| 4 | r | y | r | r | r | r | 4 |
| 5 | r | r | G | G | G | r | 31 |
| 6 | r | r | y | y | y | r | 4 |

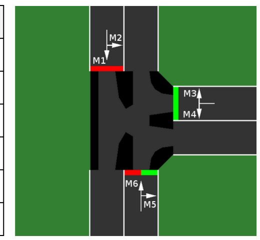



**Figure 4: Example of a semaphore plan with illustrative image for phase 5**

## 3.2 Scenario Definition

As a demonstration of the concept, it was used a grid (Manhattan-like) map (Figure 5) in order to make some experiments for traffic light control. A grid map is relatively simple to implement and where it is fairly possible to define consistent semaphore plans. The Q-learning algorithm was chosen as the learning method for the traffic light agents.

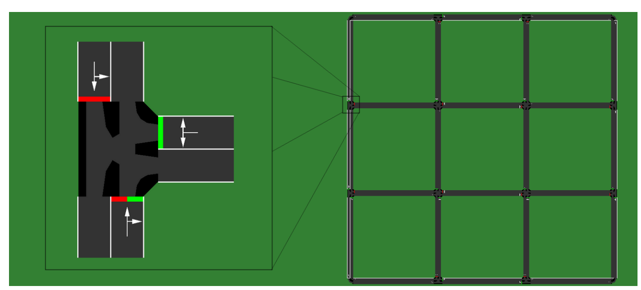


**Figure 5: The grid map where simulation took place**

Thus, these experiments consisted in performing four simulations: one with traffic lights with fixed semaphore plans, one with traffic lights with fixed semaphore plans but with different durations for distinct day periods, another with traffic lights with Q-Learning taking into account the duration of the phases and another with traffic lights with Q-Learning taking into account the duration of the phase and the period of the day. The metrics that will be used to evaluate the results are described in Section 3.3. Therefore, to be a basis for comparison, each of the simulations had the same background: the same vehicles leaving at the same time, from the same place and with the same route.

| Theoretical hour | Starting step | Traffic |
|---|---|---|
| 00h00 | 0 | Low |
| 07h30 | 150 000 | High |
| 09h00 | 180 000 | Medium |
| 18h00 | 360 000 | High |
| 20h00 | 400 000 | Low |

**Table 1: Traffic distribution during the day**

As SUMO's time unit is step (step of execution), and as each step can last more or less a second, it was necessary to make a correlation between number of steps and the time in simulation. This correlation is necessary to implement time compression and allow for an entire day to be simulated correctly and much quicker than in the real-life duration. Thus, the approach taken was that 20000 steps correspond to 1 theoretical simulation hour. There are three traffic scenarios throughout the day: low traffic, medium traffic with a predominance of horizontal flows of vehicles, and heavy traffic. The distribution of traffic is performed according to Table 1.

A manual approach was carried out for the definition of the green splitting for the phases in the simulations where Q-Learning was not used. In the specific case of the traffic lights with fixed semaphore plans but with different durations for distinct day periods, in the low traffic period faster green durations were used in opposition to the high traffic period where long green durations were used.

Each simulation corresponded to a 4-day simulation. This way, at the end of each simulation, that is, when all vehicles arrived at their destination, metrics were generated.

The tool-chain takes some time to add all desired vehicles at startup. This way, simulation time should not be such that would make the startup take longer than necessary. However, simulation time should be enough so traffic lights have time to learn. 4-day simulation seemed to be the best way for balancing these issues.

It is also important to note that the insertion of network traffic was not made in a distributed manner again because of the slowness that would result with the startup of the tool-chain. Thus, two approaches have been considered for the four simulations: on the one hand, insertions with intervals of 7000 steps, and on the other hand, insertions with intervals of 10000 steps. In each of these intervals, the quantity of vehicles to add would depend on the period of day that the simulation was on. So, in reality, there were 8 simulations.

### 3.3 A Q-Learning traffic control

It is important to be aware that the state representation has influence in the Q-Learning performance, in other words, it is only possible to learn something if it is relevant to the problem. In this sense, it is intended to use two relevant variables: phase durations and period of the day. It is considered that phases initially with duration under 20 seconds will not suffer any variation and the other phases will have durations between 20 and 60 seconds, with a granularity of 5 seconds. Assuming that could exist two or three phases with variable durations for each semaphore plan, there are a total of 81 or 729 duration combinations, respectively. Possible actions are decrease, maintain or increase (-5, 0 or +5 seconds) each variable duration, which results in a Q-Table with 243 or 2187 pairs. Considering the period of the day these numbers would increase.

The reward function consists of two portions: the own reward multiplied by $0,5$ and the weighted average (concerning distance of roads) of neighboring traffic lights rewards multiplied by $0,5$. These rewards are calculated using the average of vehicles in the vicinity of an intersection, during a complete cycle. In what concerns exploration, it is used a 0-greedy strategy. The learning rate was 50% as well as was the discount factor.

In order to evaluate the learning process, the following metrics will be used:

- Travel time and average waiting time in queues, that allow to check the individual performance of each vehicle;
- Standard deviations of travel times and of average waiting times in queues, that allow to check the network traffic homogeneity, in other words to check whether vehicles will have a similar experience both in travel time and waiting time in queues;
- Average of travel times and of average waiting times in queues, that allows to check the global network traffic performance.

### 3.4 A multi-agent system for traffic control

System could be implemented using two agent models: an agent for each traffic light with a super coordinator agent, or an agent for each traffic light with distributed coordination. First model allows a greater process synchronization between agents, has a single point of failure for the entire system and has a computation volume highly concentrated in the coordinator. The second model can hardly obtain synchronization but yet in the event of a failure, this is not spread to the entire system, and computation is homogeneous.

Therefore, system will be implemented using the second model in which agents are traffic lights. The architecture of each agent displayed in Figure 6 is based on a learning agent architecture [16, p. 54-57] but specified to the Q-Learning process. In this Figure, the presented behaviour does not include the initial phase in which the Q-Table is initialized and where each agent finds the neighbors (in Figure 6 represented as Agent n).

There exist two types of communication between agents: reward requests and answers to reward requests. The former is implemented using the performative *QUERY_REF* with content "reward", whereas the latter uses the performative *INFORM_REF* with the reward itself in the content.

Figure 7 is described a possible situation between agents, in which *Agent2* is a neighbor of *Agent1* and *Agent3*, and *Agent1* and *Agent3* are not neighbors.

## 4. PRELIMINARY RESULTS AND DISCUSSION

Figures 8 and 9 show, for each vehicle, the average waiting time in queues.

For each vehicle, with intervals of 7000 steps there are greater peaks in the average waiting time in queues compared to intervals of 10000 steps. This is explained as the network gets easily saturated with fewer steps and vehicles wait longer in queues. Another fact is that, in individual terms, the average waiting time in queues does not vary a considerably with the types of semaphore plan.

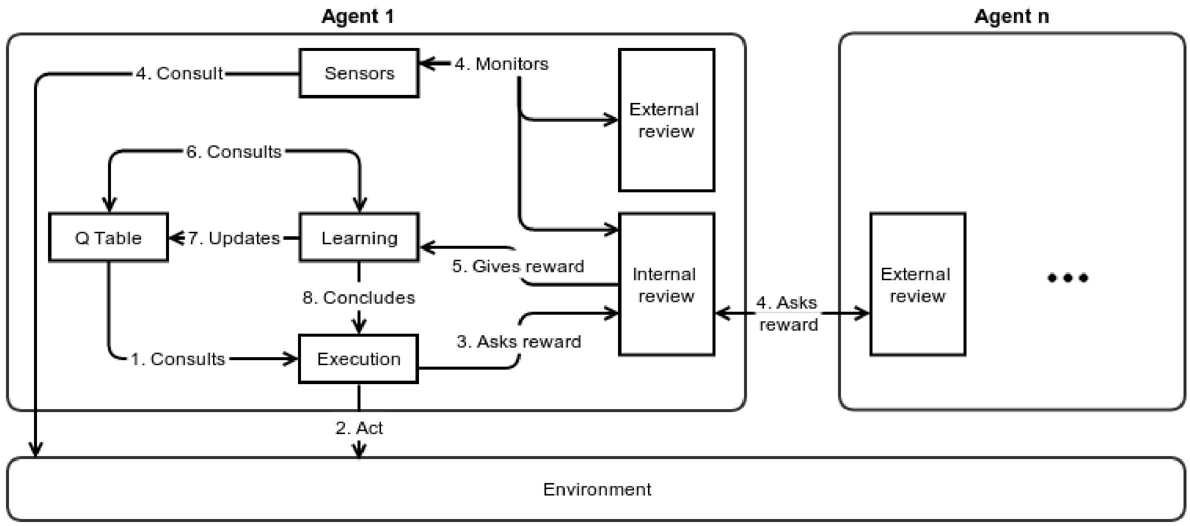



**Figure 6: Traffic light agent architecture and behaviour**

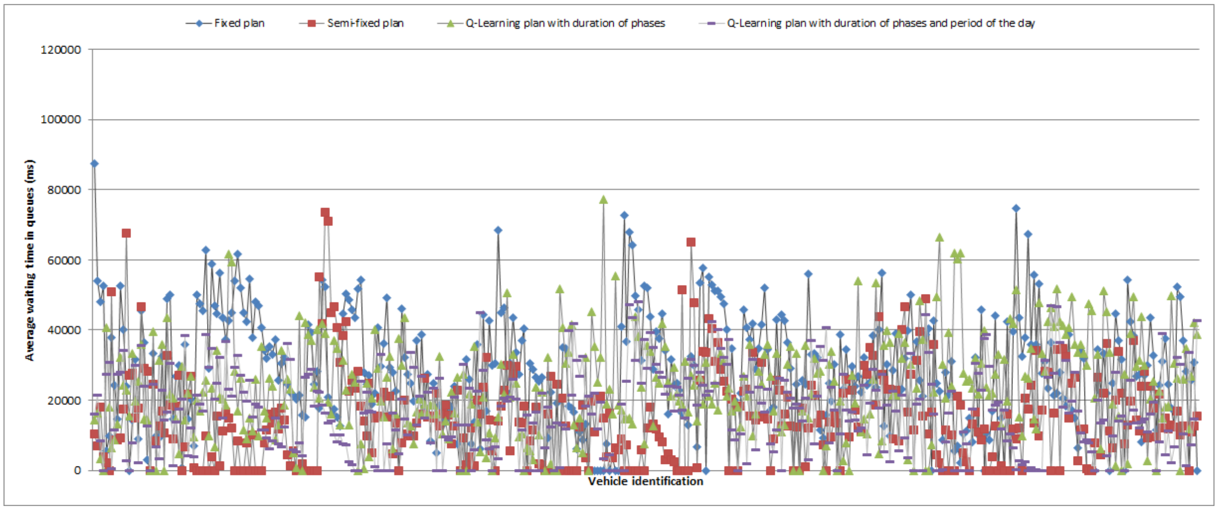



**Figure 8: The average waiting time in queues with intervals of 10000 steps**

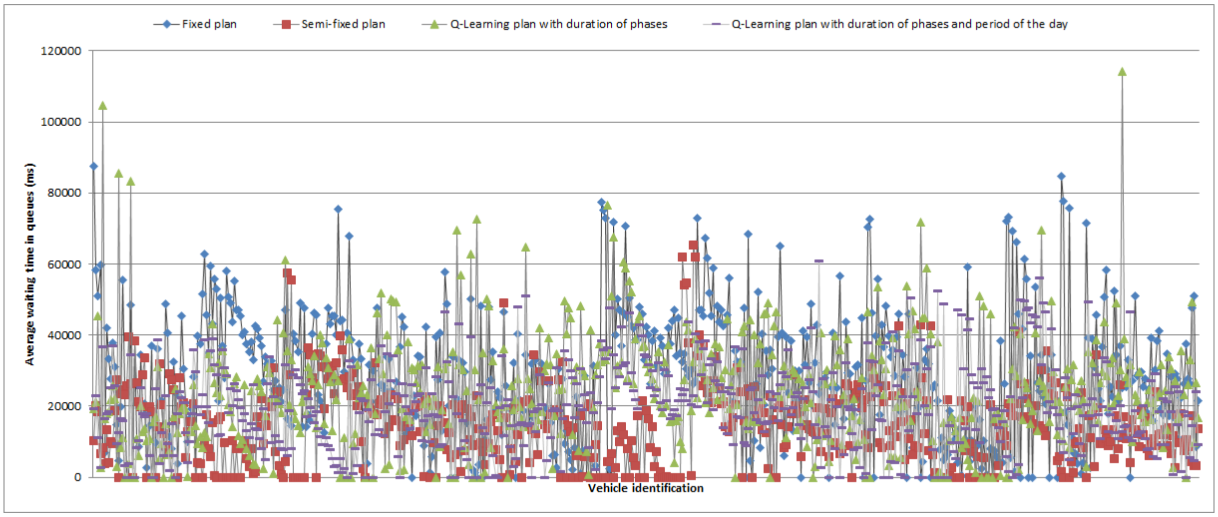



**Figure 9: The average waiting time in queues with intervals of 7000 steps**

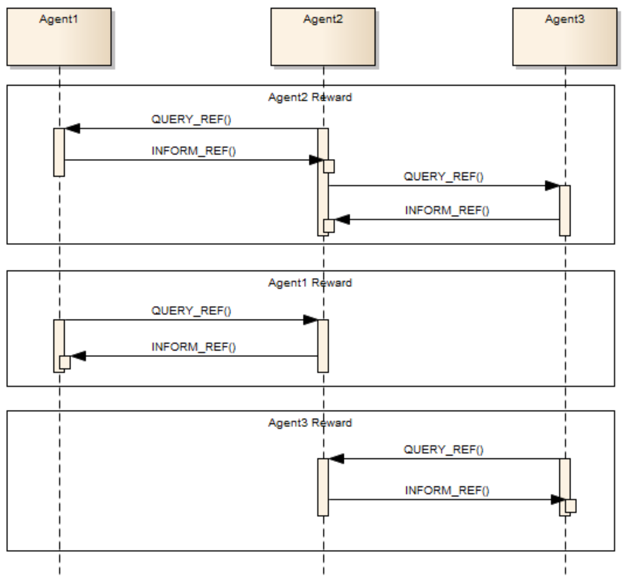



**Figure 7: Interaction example between agents**

In what concerns the travel time for each vehicle, once again intervals of 7000 steps produce greater peaks, in other words, easily a vehicle takes longer to travel the same path. It is curious to verify that with a changing of the step intervals a vehicle can take longer or shorter in different plans. In other words, unlike the previous metric, there is not a better semaphore plan for the majority of the vehicles, and so a semaphore plan can give better individual results for some vehicles, but not for all vehicles.

Figures 10 and 11 present metrics to a more global evaluation of the explored solutions. It is called *Q-Learning A* to the plan taking into account the duration of the phases and *Q-Learning B* to the plan taking into account the duration of the phases and period of the day.

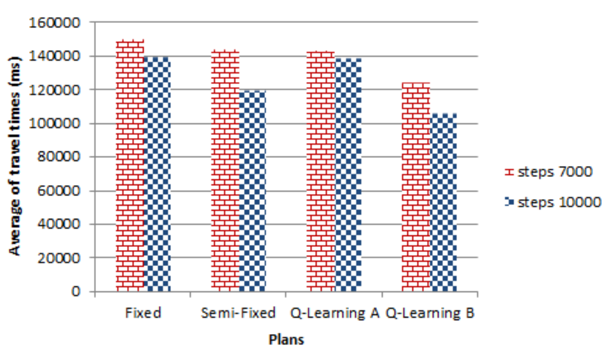



**Figure 10: Average of travel times**

In these Figures a clear difference between the fixed and semi-fixed plan is shown: while the fixed plan presents the worst results, semi-fixed plans presents the best results, even compared to the Q-Learning plans. Even so, the *Q-Learning B* plan has better results than *Q-Learning A*, as it was expected.

However, what matters the most for the driver is the total travel time. Looking at the Figures, the differences between plans are not big, mainly for the plans with intervals of 7000

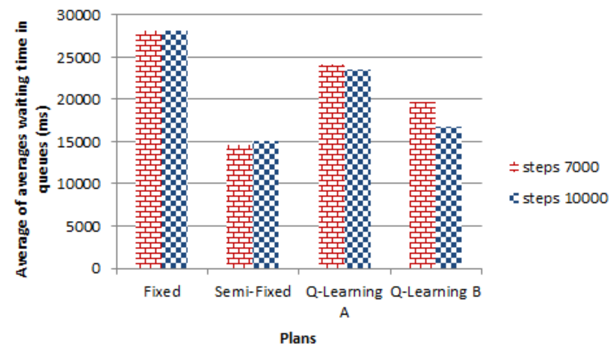



**Figure 11: Average of averages waiting time in queues**

steps. Even so, *Q-Learning B* plan has slightly better results.

The peculiar result that semi-fixed plans induces lower waiting times in queues but longer travel times than *Q-Learning B* may be explained. A simple example where this makes sense is that while in *Q-Learning B* a vehicle can pass through a lot of green traffic lights (inducing lower travel times), in the few traffic lights that it has to wait, it waits a lot of time (inducing a greater average waiting time in queues). In the semi-fixed plan a vehicle may have to wait, in average, shorter in queues but as it stops in more traffic lights than in *Q-Learning B*, it takes longer to travel through.

Finally, Figures 12 and 13 show the results of standard deviations.

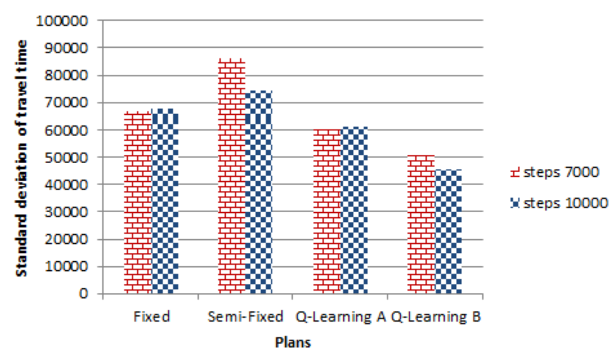



**Figure 12: Standard deviation of travel time**

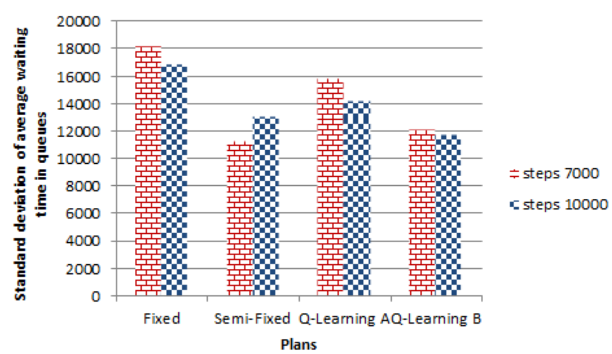



**Figure 13: Standard deviation of average waiting time in queues**

For the averages of waiting times in queues, the semi-fixed plan has the best network traffic homogeneity for intervals of 7000 steps and the second best for intervals of 10000 steps. Nevertheless, in general terms *Q-Learning B* can obtain more network traffic homogeneity.

Passing to the total travel times, *Q-Learning B* can widely overcome the other plans, obtaining greater network traffic homogeneity, both for intervals of 7000 and 10000 steps. The network traffic homogeneity is an important factor for a driver, who intends to know that when he goes to his destination there is not a probability to take longer than it was expected.

## 5. RELATED WORKS

The specific case of traffic lights is one of the areas where much has been researched for new solutions, from the design of intersections [13] (including physical layout and semaphore plans), to the definition of semaphore plans through statistical analysis. Current solutions try to answer the highly dynamic system [8, p. 343] using coordinated control. Several methodologies have been used such as genetic algorithms [18], fuzzy logic [2] and reinforcement learning [1].

To date, there are not many solutions for traffic that make full use of the intelligent agent concept. However, the multi-agent system approach has become recognized as a convenient approach for modelling and simulating complex systems [15]. Also, it has grown enormously not only applied to traffic but also to transportation in general terms [7].

In the last decade some microscopic simulators have been developed, such as MITSIM, Paramics, Aimsun, CORSIM and VisSim. However, none of these is strictly defined as agent-based simulation systems, even though they model vehicles in an object-oriented manner. Just a few simulation tools truly support the concept of agents and multi-agent systems in traffic simulation; MATSim-T [3, 4] and ITSUMO [9, 6] are good examples to be mentioned.

Regarding this simulation tools some examples of multi-agent system approaches for traffic lights control can be seen in [11], [5], [14] and [10]. Simulators used in these works were Aimsun, ITSUMO, VisSim and ITSUMO, respectively.

With MAS being recognized as a convenient approach, there must be a sufficiently general way to couple this approach to such a huge quantity of microscopic simulators that exist now. The platform that integrates SUMO and JADE consists of an API intended to allow interoperability among simulators. The platform, coined TraSMAPI, is sufficiently general to allow other simulators to interact with MAS frameworks such JADE. A previous paper [20] reports on an experiment integrating ITSUMO and SUMO under TraSMAPI, thus demonstrating such an ability. In another study [22], external traffic controller agents operate over Aimsun-simulated scenarios through TraSMAPI. In this specific work, we illustrate how non-agent-based simulators can be extended with TraSMAPI to support MAS-T assessment. There are certainly other options to simulate agent-based traffic and transportation, such as MATSim. Although such tools are open-source then allowing full customisation, the use of JADE over a traditional microscopic simulation tool is expected to promote greater flexibility in terms of agent architectures that can be implemented.

In respect to the described tool-chain, a similar approach has already been proposed. In [17] it is possible to see the tool-chain JADE+TraSMAPI+SUMO. However, the goal of this work was focused on the vehicles itself instead of traffic lights.

## 6. CONCLUSIONS

This paper explores the use of a specific tool-chain for the implementation of intelligent traffic light control. At the end, we have a tool that allows us to implement and test real MAS-based solutions in the domain of traffic and transportation, using commodity computers and open-source tools of wide reach. Q-Learning was used as the reinforcement learning method to illustrate the implementation of traffic light agents. The tool-chain resulting from the integration of JADE and SUMO through TraSMAPI is the main expected contribution of this paper.

Nonetheless, many improvements can be identified for future work. This paper did not analyse other forms for traffic control. For example, there are solutions based on the simple statistical analysis of traffic information and posterior adjustment according to such analytical procedures. This kind of solution can contrast with others as it can be highly dynamic and therefore can be applied to very specific scenarios. Another possible solution is the installation of sensors in each traffic light that, on a reactive way, can simply respond according to the number of waiting vehicles in the queue, needing neither great computation power nor the analysis of the traffic network, totally or partially.

The tool-chain itself could be improved in some different possible ways, including scalability, robustness, and efficiency. Firstly, the increase of performance in information retrieval by decreasing time in communication between the agent and the simulator. SUMO, that is still in its very young stage, proved to be much slower than desired with a larger number of vehicles and constant information retrieval. Certainly this aspect will be improved in next versions of SUMO, but it is necessary to analyse who is to blame: Is TraCI too much slow? is TraSMAPI implemented well in what concerns performance issues? Dduring simulations is the number of generated requests to TraCI greater than necessary? and so forth! On other hand, for the real simulated system implementation it would be necessary to develop a distributed system where each agent was executed in each machine.

In this specific study we did not use JADE ability to distribute agents over a computer network, as our main objective is to demonstrate how JADE and SUMO can be integrated through TraSMAPI. Nonetheless, larger networks will certainly require more robust computational power, which can be achieved through an appropriate distribution of computation across a computer network. The traffic network itself could also be improved: a more realistic map for simulation can give more relevant results. Maybe the multi-agent system used could not be the best for the proposed approach. An analysis of the best tool to use is certainly imperative.

We intend to use the proposed framework to further investigate traffic control strategies through more robust and complex signal agents. Contrary to the manual approach adopted to set up semaphore plans, tools such as Transit can be used to assist a more coherent definition of phases at each junction of the network. Finally, in terms of general results, it seems that Q-Learning taking into account the duration of the phases and the period of the day obtains better general results, even if they are not very significant. Nevertheless, it is necessary to perform these experiments in more real settings, not only in what concerns the network, but also in what concerns simulation. So, it would be possible to better conclude whether the Q-Learning implementation in traffic networks is an added value not only for drivers, but also for the system as a whole.

## 7. ACKNOWLEDGMENTS

Authors greatly acknowledge invaluable contributions by Filipe Oliveira, who also worked directly on this project. We also thank Professor Eugénio Oliveira and Dr. Henrique Lopes Cardoso for important suggestions and comments on the course of this work.

## 8. REFERENCES

[1] I. Arel, C. Liu, T. Urbanik, and A. Kohls. Reinforcement learning-based multi-agent system for network traffic signal control. *Intelligent Transport Systems, IET*, 4(2):128–135, 2010.

[2] E. Azimirad, N. Pariz, and M.-B. N. Sistani. A novel fuzzy model and control of single intersection at urban traffic network. *Systems Journal, IEEE*, 4(1):107–111, 2010.

[3] M. Balmer, K. Meister, M. Rieser, K. Nagel, K. W. Axhausen, K. W. Axhausen, and K. W. Axhausen. *Agent-based simulation of travel demand: Structure and computational performance of MATSim-T*. ETH, Eidgenössische Technische Hochschule Zürich, IVT Institut für Verkehrsplanung und Transportsysteme, 2008.

[4] M. Balmer, M. Rieser, K. Meister, D. Charypar, N. Lefebvre, K. Nagel, and K. Axhausen. MATSim-T: Architecture and simulation times. *Multi-agent systems for traffic and transportation engineering*, pages 57–78, 2009.

[5] A. L. Bazzan, D. de Oliveira, and B. C. da Silva. Learning in groups of traffic signals. *Engineering Applications of Artificial Intelligence*, 23(4):560–568, 2010.

[6] A. L. Bazzan, M. d. B. do Amarante, T. Sommer, and A. J. Benavides. ITSUMO: an agent-based simulator for its applications. In *Proc. of the 4th Workshop on Artificial Transportation Systems and Simulation. IEEE*, 2010.

[7] A. L. Bazzan and F. Klügl. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, pages 1–29, 2013.

[8] Y. K. Chin, W. Y. Kow, W. L. Khong, M. K. Tan, and K. T. K. Teo. Q-learning Traffic Signal Optimization within Multiple Intersections Traffic Network. In *Computer Modeling and Simulation (EMS), 2012 Sixth UKSim/AMSS European Symposium on*, pages 343–348. IEEE, 11 2012.

[9] B. C. da Silva, A. L. Bazzan, G. K. Andriotti, F. Lopes, and D. de Oliveira. ITSUMO: an intelligent transportation system for urban mobility. In *Innovative Internet Community Systems*, pages 224–235. Springer, 2006.

[10] D. de Oliveira, A. L. Bazzan, B. C. da Silva, E. W. Basso, L. Nunes, R. Rossetti, E. de Oliveira, R. da Silva, and L. Lamb. Reinforcement Learning based Control of Traffic Lights in Non-stationary Environments: A Case Study in a Microscopic Simulator. In *EUMAS*. Citeseer, 2006.

[11] L. B. de Oliveira and E. Camponogara. Multi-agent model predictive control of signaling split in urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 18(1):120–139, 2010.

[12] German Aerospace Center, Institute of transportation Systems. SUMO at a Glance. `http://sumo-sim.org/userdoc/Sumo_at_a_Glance.html`. Accessed: 2013-10-23.

[13] P. Koonce, L. Rodegerdts, K. Lee, S. Quayle, S. Beaird, C. Braud, J. Bonneson, P. Tarnoff, and T. Urbanik. Traffic signal timing manual. Technical report, US Department of Transportation, 2008.

[14] W. Lu, Y. Zhang, and Y. Xie. A multi-agent adaptive traffic signal control system using swarm intelligence and neuro-fuzzy reinforcement learning. In *Integrated and Sustainable Transportation System (FISTS), 2011 IEEE Forum on*, pages 233–238. IEEE, 2011.

[15] L. J. Moya and A. Tolk. Towards a taxonomy of agents and multi-agent systems. In *Proceedings of the 2007 spring simulation multiconference-Volume 2*, pages 11–18. Society for Computer Simulation International, 2007.

[16] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence. Prentice Hall, 2010.

[17] G. Soares, J. Macedo, Z. Kokkinogenis, and R. J. Rossetti. An integrated framework for multi-agent traffic simulation using sumo and jade. In *SUMO2013, The first SUMO User Conference, May 15-17, 2013 - Berlin-Adlershof, Germany*, pages 125–131. DLR - Institut für Verkehrssystemtechnik, 2013.

[18] F. Teklu, A. Sumalee, and D. Watling. A genetic algorithm approach for optimizing traffic control signals considering routing. *Computer-Aided Civil and Infrastructure Engineering*, 22(1):31–43, 2007.

[19] Telecom Italia Lab. JADE description. `http://jade.tilab.com/description-index.htm`. Accessed: 2013-10-20.

[20] I. J. Timóteo, M. R. Araújo, R. J. Rossetti, and E. C. Oliveira. Using trasmapi for the assessment of multi-agent traffic management solutions. *Progress in Artificial Intelligence*, 1(2):157–164, 2012.

[21] I. J. P. M. Timóteo, M. R. Araújo, R. J. F. Rossetti, and E. C. Oliveira. TraSMAPI: An API oriented towards Multi-Agent Systems real-time interaction with multiple Traffic Simulators. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1183–1188, 9 2010.

[22] C. Vilarinho, G. Soares, J. Macedo, J. P. Tavares, and R. J. Rossetti. Capability-enhanced {AIMSUN} with real-time signal timing control. *Procedia - Social and Behavioral Sciences*, 111(0):262 – 271, 2014. Transportation: Can we do more with less resources? - 16th Meeting of the Euro Working Group on Transportation - Porto 2013.

[23] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux. TraCI: an interface for coupling road traffic and network simulators. In *Proceedings of the 11th communications and networking simulation symposium*, pages 155–163. ACM, 2008.

# Traffic simulation with the GAMA platform

Patrick Taillandier
UMR CNRS IDEES, University of Rouen
7 rue Thomas Becket
Mont Saint Aignan, France
patrick.taillandier@univ-rouen.fr

## ABSTRACT

These last years have seen the multiplication of traffic agent-based frameworks (MATSim, SUMO...). If these frameworks are well-fitted for the study of normal traffic conditions, it is often complex to adapt them - in particular for non-computer scientists - for more specific application contexts such as the study of impacts of uncommon events (e.g. car accidents, technological hazardsĚ). In this paper, we present a new open-source (GPL) tool, integrated into the GAMA modeling and simulation platform, allowing to easily define new microscopic traffic simulations, easily tunable, with a detailed representation of the driver operational behaviors. In particular, it allows to take into account the road infrastructures and traffic signals, the change of lanes of the drivers and their respects of norms. Moreover, the tool allows to run simulations at city level with tens of thousands of driver agents. We illustrate the use of this plug-in through an example for the traffic simulation of the Rouen city (France).

## Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Distributed Artificial Intelligence

## General Terms

Design

## Keywords

Agent-based Modeling, Traffic Simulation, GAMA Platform

## 1. INTRODUCTION

Traffic simulations have proved their interests for urban planners. Many models have been developed these last years. These models are often grouped according to their levels of representation: macroscopic [9], mesoscopic [17], microscopic [12] and nanoscopic [4].

A modeling approach that is particularly well-fitted for micro-simulation is the agent-based modeling. It allows to consider the heterogeneity of the driver behaviors and to take into account the global impact of local processes. This modeling approach is in particular adapted to the study, at fine scale (spatial and temporal), of the impacts of uncommon events such as car accidents or technological hazards (see for example [16]). In this context, being able to simulate the traffic in a realistic way while taking into account the road infrastructure (crossing, traffic signals...), the properties of the cars (length, max speed...) and the personality of the drivers (tendency to respect the norms) is mandatory.

Even if there are nowadays many frameworks dedicated to the development of agent-based traffic models, many models are still developed from scratch or with a generic platform (e.g. [6, 3, 16]). Indeed, if the existing frameworks are most of the time well-fitted to the simulation of normal traffic conditions, they cannot be easily tuned by domain experts that are often not computer scientists.

In this paper, we present a new plug-in integrated in the open-source GAMA modeling and simulation platform [5] dedicated to the development of fine scale traffic simulations. GAMA provides modelers - which are not, most of the time, computer scientists - with tools to develop highly complex models. In particular, it offers a complete modeling language (GAML: GAma Modeling Language) and an integrated development environment that allows modelers to quickly and easily build models. Indeed, the GAML language is as simple to use and to understand as the Netlogo modeling language [15] and do not requires high level programming skills. The plug-in developed allows GAMA user to easily define traffic simulation at fine scale, with a detailed representation of the driver operational behaviors.

The paper is organized as follows: Section 2 presents the related works, in particular the existing agent-based traffic simulators and frameworks. Section 3 is dedicated to the presentation of the driving GAMA plug-in. Section 4 gives an example of a model developed with the plug-in. At last, Section 5 concludes and presents some perspectives.

## 2. RELATED WORKS

Many open source traffic simulation frameworks have been developed these last years.

One of the most famous is MATSim [2] (Multi Agent Transport Simulation Toolkit). MATSim is an open-source (GPL) Java application that consists of several modules which can be combined. MATSim proposes many advance features dedicated to traffic simulations that can be enrich by users through the definition of new modules in JAVA.

Another famous open-source framework is SUMO [10]. SUMO is a suite of applications which help modelers to prepare and to perform traffic simulations. Like MATSim, it proposes many advance features dedicated to traffic simulations that can be enrich using C++.

AgentPolis [7] is another open source framework dedicated to traffic simulations. In comparison to the two previously cited frameworks, AgentPolis adopts a fully agent-based modeling approach. Drivers are represented as autonomous agents with asynchronous control modules and the ability to interact freely with the environment and other agents. AgentPolis is implemented in JAVA and can be enrich using the JAVA programming language.

These three frameworks are powerful and propose many advance features. However, for modelers without high level programming skills, adapting these platforms to specific application contexts is out of reach as they require to write code in JAVA or C++.

Concerning the generic modeling and simulation platforms, only few propose tools that can be used to develop traffic simulations. One of them is Repast Symphony [13] that proposes interesting features concerning GIS loading and graphs. However, using this platform to develop complex models require to write code in JAVA. Note that for simple models, modelers can use the Relogo modeling language.

The GAMA platform [5] provides as well different features that can be used by modelers to develop traffic models. In particular, GAMA allows to simply load GIS data (shape files, OSM data...), to define graphs from polyline geometries, to compute shortest paths and to move agents on a polyline networks. If these features are well-suited for the development of traffic simulations at large time scale (see for example the MIRO project [3] - time scale: 10 minutes per step), they do not allow to simply account the driver behaviors at fine scale: his/her change of lanes, the effect of traffic signals...

In this context, we have developed a new driving plug-in for the GAMA platform. The goal is to offer to modelers a tool that is at the same time easy to use for all application contexts and that allows to build realist traffic simulations.

## 3. DRIVING GAMA PLUG-IN

### 3.1 Presentation of the plug-in

The developed tool is integrated in the GAMA platform as a plug-in. It provides modelers with new GAML instructions allowing to support the definition of traffic simulation. GAML is an agent-oriented language, in which modelers define species of agents, i.e. archetype of agents, their characteristics (variables), behaviors and aspects. The behaviors of agents are defined through actions and reflexes. An action is a block of instructions executed when called. A reflex is a block of instructions executed at each simulation step when its optional attached condition is true. An aspect represents how an agent can be displayed. The richness of GAML comes from the numerous optimized operators it integrates. In particular, GAMA provides modelers with a native integration of GIS data and allows to easily load shapefiles, OSM data and to use databases. It integrates as well many graph operators.

Concerning our tool, we chose to represent all the road infrastructures (road, traffic signalsĚ) as agents. The main interest of this is to give the modelers the possibility to simply add dynamics to these infrastructures: e.g. to add a deterioration dynamic to roads.

Our tool takes the form of three GAMA skills. A skill is a built-in module that provides a set of related built-in variables and built-in actions (programmed in JAVA) to the species of agents that declare them. In particular, we define 3 new skills:

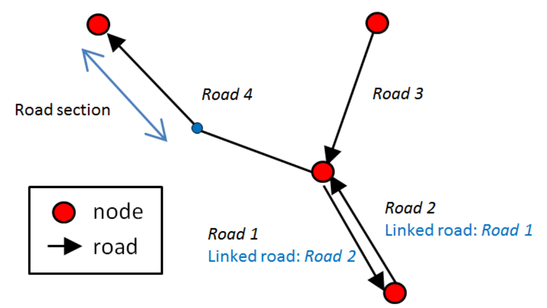



**Figure 1: Roads and nodes**

- *Advanced driving skill*: dedicated to the definition of the driver species. It provides the driver agents with variables and actions allowing to move an agent on a graph network and to tune its behavior.
- *Road skill*: dedicated to the definition of roads. It provides the road agents with variables and actions allowing to registers agents on the road.
- *RoadNode Skill*: dedicated to the definition of node. It provides the node agents with variables allowing to take into account the intersection of roads and the traffic signals.

### 3.2 Structure of the network: road and roadNode skills

A key issue for our tool is to be versatile enough to be usable with most of classic road GIS data, in particular OSM data. We choose then to use a classic format for the roads and nodes (See Figure 1). Each road is a polyline composed of road sections (segments). Each road has a target node and a source node. Each node knows all its input and output roads. A road is considered as directed. For bidirectional roads, 2 roads have to be defined corresponding to both directions. Each road will be the *linked_road* of the other. Note that for some GIS data, only one road is defined for bidirectional roads, and the nodes are not explicitly defined. In this case, it is very easy, using the GAML language, to create the reverse roads and the corresponding nodes (it only requires few lines of GAML).

A lane can be composed of several lanes (Figure 2) and the vehicles will be able to change at any time its lane. Another property of the road that will be taken into account is the maximal authorized speed on it. Note that even if the user of the plug-in has no information about these values for some of the roads (the OSM data are often incomplete), it is very easy using the GAML language to fill the missing value by a default value. It is also possible to change these values dynamically during the simulation (for example, to take into account that after an accident, a lane of a road is closed or that the speed of a road is decreased by the authorities).

The *road skill* provides the road agents with several variables that will define the road properties:

- *lanes*: integer, number of lanes.

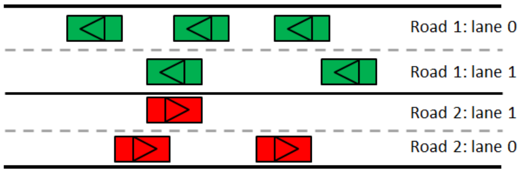



**Figure 2: Roads and lanes**

- *maxspeed*: float point value; maximal authorized speed on the road.
- *linked_road*: road agent; reverse road (if there is one).
- *source_node*: node agent; source node of the road.
- *target_node*: node agent; target node of the road.

It provides as well the road agents with one read only variable:

- *agents_on*: list of list (of driver agents); for each lane, the list of driver agents on the road.

The *road skill* provides the road agents with several variables that will define the road properties:

- *roads_in*: list of road agents; the list of road agents that have this node for target node.
- *roads_out*: list of road agents; the list of road agents that have this node for source node.
- *stop*: list of list of road agents; list of stop signals, and for each stop signal, the list of concerned roads.

It provides as well the road agents with one read only variable:

- *block*: dictionary (map): key: driver agent, value: list of road agents; the list of driver agents blocking the node, and for each agent, the list of concerned roads.

### 3.3 Advanced driving skill

Concerning the driver agents, we propose a driving model based on the one proposed by [16]. In the model proposed by [16], each driver agent has a planned trajectory that consists in a succession of edges. When the driver agent enters a new edge, it first chooses its lane according to the traffic density, with a bias for the rightmost lane. The movement on an edge is inspired by the Intelligent Driver Model [8]. A difference with our driving model is that in our model the drivers have the possibility to change their lane at any time (and not only when entering a new edge). In addition, we have defined more variables for the driver agents in order to give more possibilities for the modelers to tune the driver behavior.

The *advanced driving skill* provides the driver agents with several variables that will define the car properties and the personality of the driver:

- *final_target*: point; final location that the agent wants to reach (its goal).
- *vehicle_length*: float point value; length of the vehicle.
- *max_acceleration*: float point value; maximal acceleration of the vehicle.
- *max_speed*: float point value; maximal speed of the vehicle.
- *right_side_driving*: boolean; do drivers drive on the right side of the road?
- *speed_coef*: float point value; coefficient that defines if the driver will try to drive above or below the speed limits.
- *security_distance_coeff*: float point value; coefficient for the security distance. The security distance will depend on the driver speed and on this coefficient.
- *proba_lane_change_up*: float point value; probability to change lane to a upper lane if necessary (and if possible).
- *proba_lane_change_down*: float point value; probability to change lane to a lower lane if necessary (and if possible).
- *proba_use_linked_road*: float point value; probability to take the reverse road if necessary (if there is a reverse road).
- *proba_respect_priorities*: float point value; probability to respect left/right (according to the driving side) priority at intersections.
- *proba_respect_stops*: list of float point values; probabilities to respect each type of stop signals (traffic light, stop sign...).
- *proba_block_node*: float point value; probability to accept to block the intersecting roads to enter a new road.

It provides as well the driver agents with several read only variables:

- *speed*: float point value; speed expected according to the road *max_value*, the car properties, the personality of the driver and its *real_speed* (see Equation 1 for more details).
- *real_speed*: float point value; real speed of the car (that takes into account the other drivers and the traffic signals).
- *current_path*: path (list of roads to follow); the path that the agent is currently following.
- *current_target*: point; the next target to reach (sub-goal). It corresponds to a node.
- *targets*: list of points; list of locations (sub-goals) to reach the final target.
- *current_index*: integer; the index of the current goal the agent has to reach.
- *on_linked_road*: boolean; is the agent on the linked road?

Of course, the values of these variables can be modified at any time during the simulation. For example, the probability to take a reverse road (*proba_use_linked_road*) can be increased if the driver is stucked for several minutes behind a slow vehicle.

In addition, the *advanced driving skill* provides the driver agents with several actions:

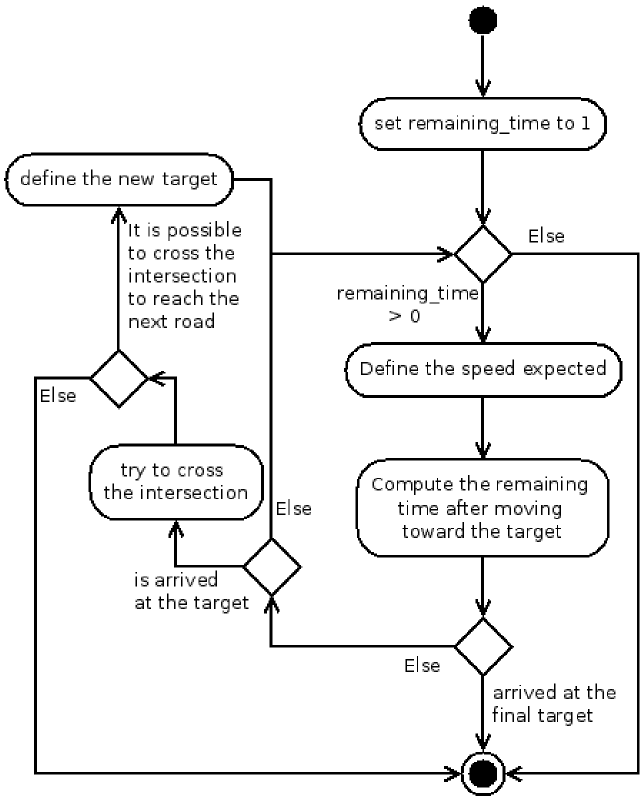



**Figure 3: Drive action**

- *compute_path*: arguments: a graph and a target node. This action computes from a graph the shortest path to reach a given node.
- *drive*: no argument. This action moves the driver on its current path according to the traffic condition and the driver properties (vehicle properties and driver personality).

the *drive* action works as follow (Figure 3): while the agent has the time to move ($remaining\_time > 0$), it first defines the speed expected. This speed is computed from the *max_speed* of the road, the current *real_speed*, the *max_speed*, the *max_acceleration* and the *speed_coef* of the driver (see Equation 1). Then, the agent moves toward the current target and compute the remaining time. During the movement, the agents can change lanes (see below). If the agent reaches its final target, it stops; if it reaches its current target (that is not the final target), it tests if it can cross the intersection to reach the next road of the current path. If it is possible, it defines its new target (target node of the next road) and continues to move.

$$\begin{aligned} speed_{driver} = Min(max\_speed_{driver}, \\ Min(real\_speed_{driver} + max\_acceleration_{driver}, \\ max\_speed_{road} * speed\_coef_{driver})) \end{aligned} \tag{1}$$

The function that defines if the agent crosses or not the intersection to continue to move works as follow (Figure 4): first, it tests if the road is blocked by a driver at the intersection (if the road is blocked, the agent does not cross the

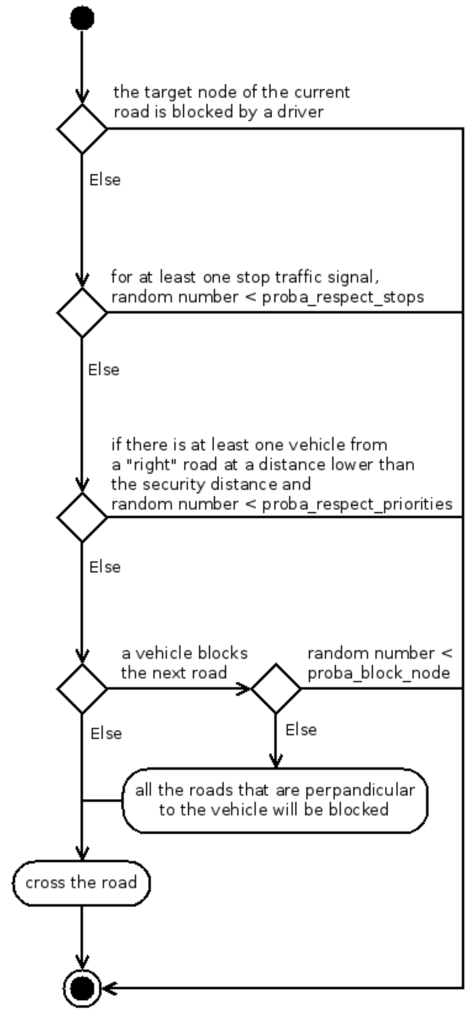



**Figure 4: Crossing of an intersection (case where *right_side_driving* is true)**

intersection). Then, if there is at least one stop signal at the intersection (traffic signal, stop sign...), for each of these signals, the agent tests its probability to respect or not the signal (note that the agent has a specific probability to respect each type of signals). If there is no stopping signal or if the agent does not respect it, the agent checks if there is at least one vehicle coming from a right (or left if the agent drives on the left side) road at a distance lower than its security distance. If there is one, it tests its probability to respect this priority. If there is no vehicle from the right roads or if it chooses to do not respect the right priority, it tests if it is possible to cross the intersection to its target road without blocking the intersection (i.e. if there is enough space in the target road). If it can cross the intersection, it crosses it; otherwise, it tests its probability to block the node: if the agent decides nevertheless to cross the intersection, then the perpendicular roads will be blocked at the intersection level (these roads will be unblocked when the agent is going to move).

Concerning the movement of the driver agents on the current road (Figure 5), the agent moves from a section of the road (i.e. segment composing the polyline) to another sec-

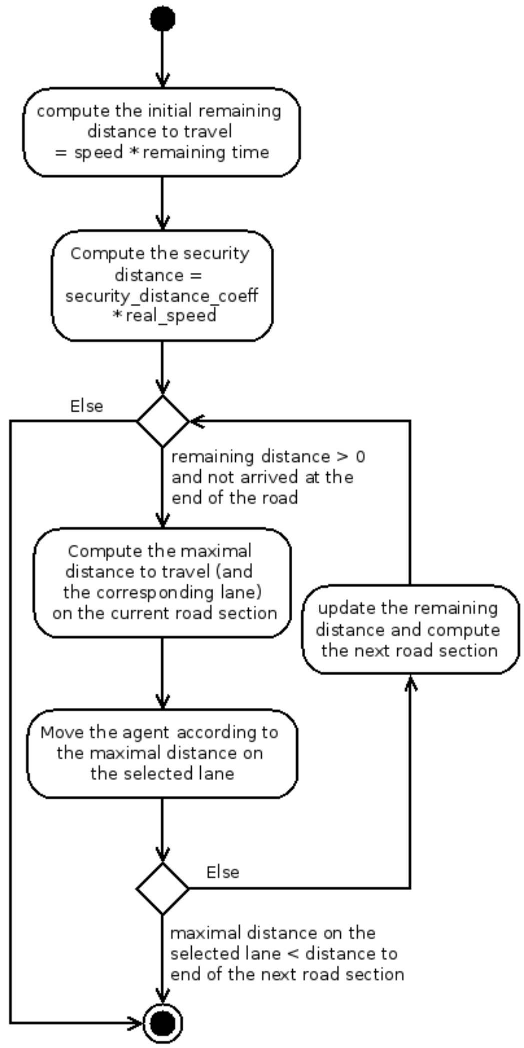



**Figure 5: Move on the current road**

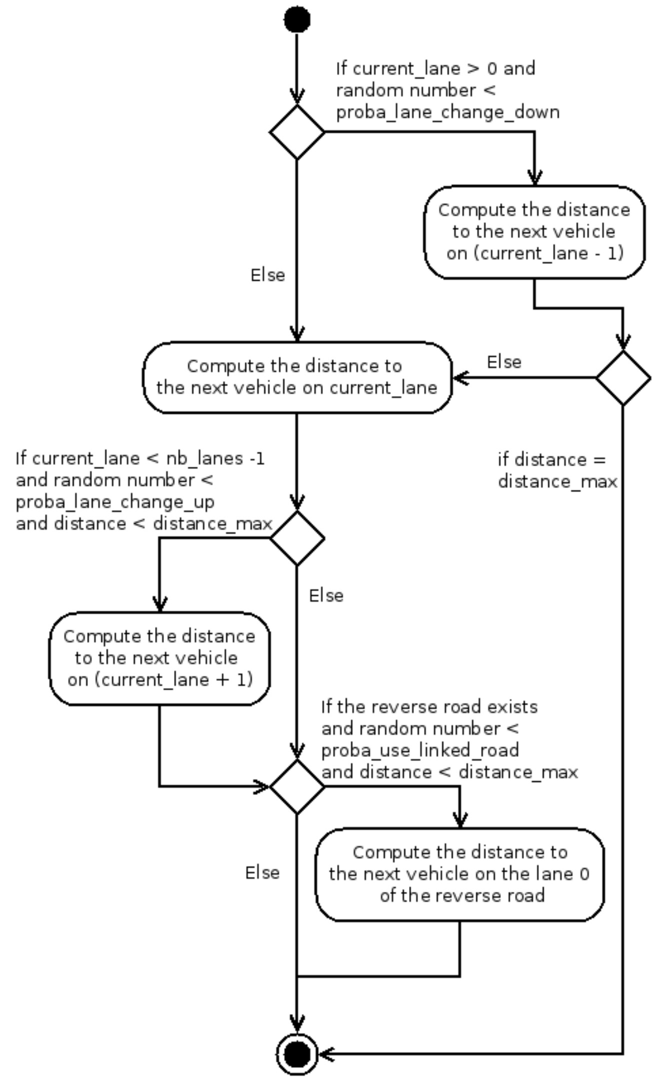



**Figure 6: Define the maximal distance possible to travel and the corresponding lane (case where *right_side_driving* is true)**

tion according to the maximal distance that the agent can moves (that will depend on the remaining time). For each road section, the agent first computes the maximal distance it can travel according the remaining time and its speed. Then, the agent computes its security distance according to its speed and its *security_distance_coeff*. While its remaining distance is not null, the agent computes the maximal distance it can travel (and the corresponding lane), then it moves according to this distance (and update its current lane if necessary). If the agent is not blocked by another vehicle and can reach the end of the road section, it updates its current road section and continues to move.

The computation of the maximal distance an agent can move on a road section consists in computing for each possible lane the maximal distance the agent can move. First, if there is a lower lane, the agent tests the probability to change its lane to a lower one. If it decides to test the lower lane, the agent computes the distance to the next vehicle on this lane and memorizes it. If this distance corresponds to the maximal distance it can travel, it chooses this lane; otherwise it computes the distance to the next vehicle on its current lane and memorizes it if it is higher than the current memorized maximal distance. Then if the memorized distance is lower than the maximal distance the agent can travel and if there is an upper lane, the agents tests the probability to change its lane to a upper one. If it decides to test the upper lane, the agent computes the distance to the next vehicle on this lane and memorizes it if it is higher than the current memorized maximal distance. At last, if the memorized distance is still lower than the maximal distance it can travel, if the agent is on the highest lane and if there is a reverse road, the agent tests the probability to use the reverse road (linked road). If it decides to use the reverse road, the agent computes the distance to the next vehicle on the lane 0 of this road and memorizes the distance if it is higher than the current memorized maximal distance.

## 3.4 Discussion

As presented above, the plug-in allows to simplify the work of modelers for the definition of traffic simulations with the GAMA platform. Of course, the plug-in does not make GAMA as rich as the existing frameworks for the development of such simulations. In particular, it proposes no tools for the pre-processing of data and do not propose any features concerning the definition of the construction of the driver daily activities. However, our tool is perfectly adapted to modelers that are not computer scientists and that want to quickly create a specific traffic model (or at least a prototype) that is not possible to create using directly the existing framework. The success of generic and simple platforms such as Netlogo [15] or GAMA have proved the interest of researchers from many research fields (geographers, sociologists...) for this kind of tools.

# 4. APPLICATION EXAMPLE

We illustrate the use of our plug-in for a simple model concerning the simulation of the traffic of the city of Rouen (France, Normandie). This city of 111553 inhabitants is built on the two sides of the Seine River. Five bridges allow to cross the river. These bridges are particularly critical for the traffic in Rouen. For instance, a truck (transporting fuel) accident has caused the closing of the Mathilde bridge since the 29th October of 2012. This bridge, which was the most used to cross the Seine (80 000 vehicles per day), should remain closed until summer 2014. This accident had (and still have) an important impact on the traffic as it has led to the multiplication of traffic jams.

As the goal of this model is just to illustrate the use of the new driving plug-in, we did not use real data to define the driver origin and destination: we affected to each driver agent a random initial location (one of the node) and a random final target (one of the node). When a driver agent reaches its destination, it just chooses a new random final target. In the same way, we did not define any specific behavior to avoid traffic jam for the driver agents: once they compute their path (all the driver agents use for that the same road graph with the same weights), they never re-compute it even if they are stucked in a traffic jam. Concerning the traffic signals, we just consider the traffic lights (without any pre-processing: we consider the raw OSM data). One step of the simulation represents 1 second. At last, in order to clarify the explanation of the model, we chose to do not present the parts of the GAML code that concern the simulation visualization. The complete model is available on the GAMA SVN (is downladable from the GAMA website [1]).

Figure 7 shows the total area (road and node shapefiles) that we choose to take into account in the simulation. This area is composed of 8000 roads and 6000 nodes. We used the OSM data (converted as shapefiles) of Rouen. A pre-process has been applied on the data in order to create a node shapefile from the road shapefile: a node is created at the extremity of each road (when several roads intersect each other, only one node is created at the intersection). Note that a GAMA model, available on the GAMA SVN, allows to directly pre-process the OSM data and to create the node and road shapefiles from them.

The following code shows the definition of species to represent the road infrastructure:

```
species road skills: [skill_road] {
  string oneway;
}
```

**Figure 7: Total area simulated: in black the roads and in yellow the nodes**

```
species node skills: [skill_road_node] {
  bool is_traffic_signal;
  int time_to_change <- 100;
  int counter <- rnd (time_to_change) ;

  reflex dynamic when: is_traffic_signal {
    counter <- counter + 1;
    if (counter >= time_to_change) {
      counter <- 0;
      stop[0] <-empty(stop[0])? roads_in : [];
    }
  }
}
```

In order to use our driving plug-in, we just have to add the *skill_road_node* to the *node* species and the *skill_road* to the *road* species. In addition, we added to the road species a variable called *oneway* that will be initialized from the OSM data and that represents the traffic direction (see the OSM map features for more details). Concerning the node, we defined 3 new attributes:

- *is_traffic_signal*: boolean; is the node a traffic light?
- *time_to_change*: integer; represents for the traffic lights the time to pass from the red light to the green light (and vice versa).
- *counter*: integer; number of simulation steps since the last change of light color (used by the traffic light nodes).

In addition, we defined for the *node* species a reflex (behavior) called *dynamic* that will be activated only for traffic light nodes and that will increment the *counter* value. If this counter is higher than *time_to_change*, this variable is set to 0, and the node change the value of the *stop* variable: if the traffic light was green (i.e. there is no road concerns by this stop sign), the list of block roads is set by all the roads that enter the node; if the traffic light was red (i.e. there is at least one road concerns by this stop sign), the list of block roads is set to an empty list.

The following code shows the definition of driver species:

```
species driver skills: [advanced_driving] {
  reflex time_to_go when: final_target = nil {
    current_path <- compute_path(
        graph: road_network, target: one_of(node));
  }
  reflex move when: final_target != nil {
    do drive;
  }
}
```

In order to use our driving plug-in, we just have to add the *advanced_driving* to the *driver* species. For this species, we defined two reflexes:

- *time_to_go*: activated when the agent has no final target. In this reflex, the agent will randomly choose one of the nodes as its final target, and computed the path to reach this target using the *road_network* graph. Note that it will have been possible to take into account the knowledge that each agent has concerning the road network by defining a new variable of type map (dictionary) containing for each road a given weight that will reflect the driver knowledge concerning the network (for example, the known traffic jams, its favorite roads....) and to use this map for the path computation.

- *move*: activated when the agent has a final target. In this reflex, the agent will drive in direction of its final target.

We describe in the following code how we initialize the simulation:

```
init {
  create node from: file("nodes.shp") with:[
    is_traffic_signal::read("type")="traffic_signals"];

  create road from: file("roads.shp")
    with:[lanes::int(read("lanes")),
    maxspeed::float(read("maxspeed")),
    oneway::string(read("oneway"))]
    {
      switch oneway {
        match "no" {
          create road {
            lanes <- myself.lanes;
            shape <- polyline(reverse
              (myself.shape.points));
            maxspeed <- myself.maxspeed;
            linked_road <- myself;
            myself.linked_road <- self;
          }
        }
        match "-1" {
          shape <- polyline(reverse(shape.points));
        }
      }
    }
  }
  map general_speed_map <-  road as_map
    (each::(each.shape.perimeter/(each.maxspeed)));

  road_network <-  (as_driving_graph(road, node))
     with_weights general_speed_map;

  create driver number: 100000 {
    location <- one_of(node).location;
    vehicle_length <- 3.0;
    max_acceleration <- 0.5 + rnd(500) / 1000;
    speed_coeff <- 1.2 - (rnd(400) / 1000);
    right_side_driving <- true;
    proba_lane_change_up <- rnd(500) / 500;
    proba_lane_change_down <- 0.5+ (rnd(250) / 500);
    security_distance_coeff <- 3 - rnd(2000) / 1000);
    proba_respect_priorities <- 1.0 - rnd(200/1000);
    proba_respect_stops <- [1.0 - rnd(2) / 1000];
    proba_block_node <- rnd(3) / 1000;
    proba_use_linked_road <- rnd(10) / 1000;
  }
}
```

In this code, we create the node agents from the node shapefile (while reading the attributes contained in the shapefile), then we create in the same way the road agents. However, for the road agents, we use the *oneway* variable to define if we should or not reverse their geometry (*oneway* = "-1") or create a reverse road (*oneway* = "no"). Then, from the road and node agents, we create a graph (while taking into account the *maxspeed* of the road for the weights of the edges). This graph is the one that will be used by all agents to compute their path to their final target. Finally, we create 10000 driver agents. At initialization, they are randomly placed on the nodes; their vehicle has a length of 3m; the maximal acceleration of their vehicle is randomly drawn between 0.5 and 1; the speed coefficient of the driver is randomly drawn between 0.8 and 1.2; they are driving on the right side of the road; their probability of changing lane for a upper lane is randomly drawn between 0 and 1.0; their probability of changing lane for a lower lane is randomly drawn between 0.5 and 1.0; the security distance coefficient is randomly drawn between 1 and 3; their probability to respect priorities is randomly drawn between 0.8 and 1; their probability to respect light signal is randomly drawn between 0.998 and 1; their probability to block a node is randomly drawn between 0 and 0.003; their probability to use the reverse road is randomly drawn between 0 and 0.01;

We carried out a simulation of 1000 simulation steps (1000 seconds) on a i7 computer using only one of the computer cores. The duration of the simulation (without taking into account the time taken by the displaying of the simulation) was 1 second per step if we take into account the time spent by the shortest path computation by the Dijkstra algorithm or 0.3 second per step if we do not. Figure 8 shows a snapshot of the simulation. We can observe the emergence of traffic jams, driver agents stopping at a red traffic light and using the different lanes of the roads.

## 5. CONCLUSION

In this paper, we presented a new plug-in for the GAMA platform dedicated to the development of traffic simulations. This plug-in allows to define new traffic simulations with a detailed representation of the driver operational behaviors.

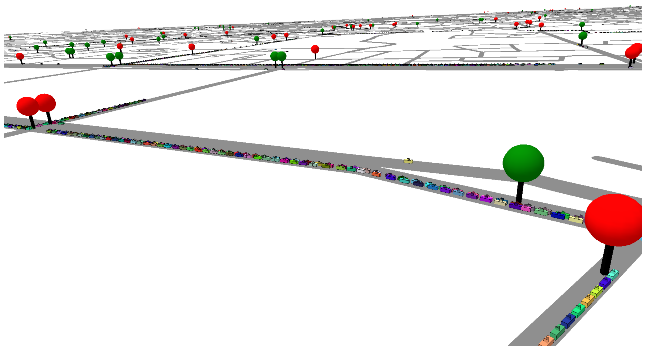


**Figure 8: Snapshot of the simulation (simulation step: 1000): The driver agents are represented by the rectangles with a triangle on top; the traffic lights are represented by the sticks with a red/green sphere on top**

In particular, it allows to take into account the road infrastructures and traffic signals, the change of lanes of the drivers and their respect of norms. We illustrated the use of our plug-in by a simple model concerning the simulation of the traffic of the city of Rouen.

In comparison to existing traffic simulation frameworks, the advantage of our tool is to enable modelers to easily define models adapted to their application context. Indeed, the use of the GAML language enables modelers without high-level programming skills to develop their own models or at least prototypes.

If the plug-in allows yet to simulate tens of thousands of driver agents, we plan to improve its efficiency by using High Performance Computing and in particular distribution on GPU to enable to carry out large scale simulation with millions of driver agents.

In addition, we plan to enrich the driving skill in order to make the driver agents more cognitive, in particular concerning their choice of path and their adaptation to the their current context. For this, we plan to give the driver agents a BDI architecture that can be based on [14, 11].

As last, we plan as well to develop new tools to help people to prepare their data. The goal will be to offer the possibility from incomplete OSM data (OSM are often incomplete) to automatically fill the missing attributes, and to create a consistent network (with its infrastructure and traffic signals). A particular attention will be brought on traffic signals and traffic lights.

# 6. REFERENCES


[1] Gama website: https://code.google.com/p/gama-platform, January 2014.

[2] M. Balmer, M. Rieser, K. Meister, D. Charypar, N. Lefebvre, K. Nagel, and K. Axhausen. Matsim-t: Architecture and simulation times. *Multi-Agent Systems for Traffic and Transportation Engineering*, pages 57–78, 2009.

[3] A. Banos, N. Marilleau, and M. Team. Improving individual accessibility to the city: an agent-based modelling approach. In *ECSS*, 2012.

[4] N. Daiheng. 2dsim: A prototype of nanoscopic traffic simulation. In *Intelligent Vehicles Symposium*, pages 47–52, 2003.

[5] A. Grignard, P. Taillandier, B. Gaudou, D. Vo, N. Huynh, and A. Drogoul. Gama 1.6: Advancing the art of complex agent-based modeling and simulation. In *PRIMA 2013: Principles and Practice of Multi-Agent Systems*, volume 8291 of *Lecture Notes in Computer Science*, pages 117–131, 2013.

[6] M. Horn. Multi-modal and demand-responsive passenger transport systems: a modelling framework with embedded control systems. *Transportation Research Part A: Policy and Practice*, 36(2):167–188, 2002.

[7] M. Jakob and Z. Moler. Modular framework for simulation modelling of interaction-rich transport systems. In *Proceedings of the 16th IEEE Intelligent Transportation Systems Conference (ITSC 2013)*, 2013.

[8] A. Kesting, M. Treiber, and D. Helbing. General lane-changing model mobil for car-following models. *Journal of the Transportation Research Board*, 1999:86–94, 2007.

[9] A. Kotsialos, M. Papageorgioy, C. Diakaki, Y. Pavlis, and F. Middleham. Traffic flow modeling of large-scale motor- way networks using the macroscopic modeling tool metanet. *IEEE Transactions on Intelligent Transportation Systems*, 3(4):282–292, 2002.

[10] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, 2012.

[11] V. M. Le, B. Gaudou, P. Taillandier, and D. A. Vo. A new bdi architecture to formalize cognitive agent behaviors into simulations. In *Advanced Methods and Technologies for Agent and Multi-Agent Systems (KES-AMSTA)*, volume 252 of *Frontiers in Artificial Intelligence and Applications*, pages 395–403. IOS Press, 2013.

[12] J. E. Miller, D. J. Hunt, J. E. Abraham, and S. P. A. Microsimulating urban systems. *Computers, Environment and Urban Systems*, 28:9–44, 2004.

[13] M. North, N. Collier, J. Ozik, E. Tatara, C. Macal, M. Bragen, and P. Sydelko. Complex adaptive systems modeling with repast simphony. *Complex Adaptive Systems Modeling*, 1(1):3, 2013.

[14] P. Taillandier, O. Therond, and B. Gaudou. A new bdi agent architecture based on the belief theory. application to the modelling of cropping plan decision-making. In *International Environmental Modelling and Software Society (iEMSs)*, 2012.

[15] S. Tisue and U. Wilensky. Netlogo: A simple environment for modeling complexity. In *International Conference on Complex Systems*, pages 16–21, 2004.

[16] P. Tranouez, E. Daudé, and P. Langlois. A multiagent urban traffic simulation. *Journal of Nonlinear Systems and Applications*, 3(2):98–106, 2012.

[17] K. Waldeer. Numerical investigation of a mesoscopic vehicular traffic flow model based on a stochastic acceler- ation process. *Transport Theory and Statistical Physics*, 33(1):31–46, 2004.

# Enhancement of Airport Collaborative Decision Making through Applying Agent System with Matching Theory

Antonio C. de Arruda Junior
TransLab, CIC, CP 4466
University of Brasilia – UnB
Brasilia-DF, 70910-900, Brazil
jnarrd@yahoo.com.br

Li Weigang
TransLab, CIC, CP4466
University of Brasilia – UnB
Brasilia-DF, 70910-900, Brazil
weigang@unb.br

Kamila B. Nogueira
Ministry of Education
Esplanada dos Ministérios,
Brasília-DF, 70047-900, Brazil
kamila.b.nogueira@gmail.com

## ABSTRACT
The Collaborative Decision Making (CDM) paradigm attempts to improve the exchange of information among the various stakeholders involved in Air Traffic Management (ATM). It is aimed at efficient decision making in airport management. Although the processes of CDM are considered mature and well accepted, in many cases it usually the focus is on the information sharing and is still not able to simultaneously involve essential agents such as Air Traffic Control (ATC) agency, airlines, and airport managers in the decision making. This study uses the matching approach of Game Theory to construct a two-sided matching market model for slot allocation in the Compression step while taking into account Ground Delay Programs (GDP). Our proposed model, Deferred Acceptance CDM (DA-CDM), assigns each flight to each slot through a "one-to-one" relationship, respecting the preferences of each allocation, leading to a stable result. It is applied to evaluate the classic CDM and Airport CDM processes with a group of analytics data. Our results show that the new allocation mechanism provides a stable and satisfactory matching of the flights with the slots in A-CDM procedure.

## Categories and Subject Descriptors
I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence – Intelligent agents, Multiagent systems; I.6.5 Computing Methodologies, Simulation and Modeling, Model Development.

## General Terms
Algorithms, Management, Design, Theory

## Keywords
Multiagent Systems, Collaborative Decision Making, Ground Delay Program, Matching Theory.

## 1. INTRODUCTION
Over the last few years, the increasing global demand for air transportation has greatly increased the complexity of the air traffic management scenario [5]. This situation enforces new integration challenges faced by several stakeholders, such as regulation agents, airlines, airport management companies, traffic managers, flight crew, passengers, and aeronautical system's manufacturers, among others [17].

Some processes, such as those aimed at reducing congestion in specific locations in the air scenario, involves the definition of delays for aircrafts on ground and are carried through the Ground Delay Program (GDP). This process, based on Collaborative Decision Making (CDM) concepts, brings the need of reallocating aircraft from the scheduled slots originally established for the runways of the affected airports [27].

Besides its simplicity of concepts, the current CDM model involves a limited number of entities in the decision-making process [6]. When using traditional CDM model and considering the existence of distinct interests on delays applied to aircraft, it is difficult task to get the satisfaction of all stakeholders who affect and are affected by delays generated by a GDP [23].

In this context, the matching approach of Game Theory can be used to construct the model of markets with the satisfactory results regarding the dispute for resources. By this approach, the preferences of all participants in that market are taken into account [25].

Regardless of the application area, a market can be modeled in order to obtain results that account for the different goals multiple agents, such as students, schools, doctors, hospitals, patients, passengers, airlines, and airports, among others. Moreover, the modeling constraints on organ donation markets in the 2000s allowed the correct treatment of a wide variety of features in more complex scenarios [21, 22].

In situations involving the departure coordination, traffic, and arrival of multiple flights through Air Traffic Management (ATM), mathematicians, economists, engineers, computer scientists, and researchers from various fields have developed Artificial Intelligence, multi-agent systems, and models based on Game Theory, among others. These models are applied in domains that involve problems of coordination and competition for resources [1, 3, 9, 28, 29].

However, most of the studies dealing with problems regarding GDP take into account only the interests of traffic control institutions and airlines. The limitation of these works based on the classic CDM model might lead to a limited level of satisfaction among other agents in the CDM process, and, consequently, the results of the process may not be stable [23].

In Brazil, this fact can be verified by the current situation, in which several concessionaires formed by private companies are entering the market to manage the major airports of the country [15]. The project, which aims to improve the quality of services and airport infrastructure, enlarging the supply of air transport to Brazilian population, currently handles billions of *reais* (Brazil's currency) and has duration of 20 to 30 years, depending on the granting rules.

Although the role of the airport operators is of crucial significance, the ATM process currently only accounts for the ATC agency. Airlines and airport managers still do not participate in the decision-making process. Also, it lacks methods to model the association of these partners in A-CDM, as well as the evaluation of distinct objectives between participating private and public companies. In this context, our main contribution is the design of a new model named Deferred Acceptance – CDM (DA-CDM), using the matching approach of Game Theory. This latter approach allows the expansion of the concepts defined in the classic CDM to more general cases. The participation of the decision making with airlines and airport managers is modeled as two-sided markets. By relying on the Deferred Acceptance algorithm [12], a stable output is guaranteed as this algorithm ensures the proper treatment of various goals amongst agents in the process of relocation of slots in a GDP. With the application of the developed algorithm and by comparison to the Compression algorithm, we are able to show satisfactory matching of the flights with the available slots in the A-CDM procedure.

The remainder of the paper is organized as follows. In Section 2 we discuss the related studies on collaborative decision making and matching markets. The slots allocation algorithms based on the classic CDM are presented in section 3. Section 4 describes the proposed DA-CDM model, and section 5 presents the evaluation of our proposal through a comparative analysis of models. We conclude in Section 6 and give suggestions for future work.

## 2. RELATED WORK

To ensure the safety and flow of flights, ATM deals with the possible inequality between demand for airspace use and capacity of the existing aviation and airport infrastructure [7]. On the other hand, ATM is considered an extremely complex and highly specialized task, besides being strongly based on the experience of the traffic manager. Its activities address critical issues such as efficiency (fluency and delays reduction), equity (working with different airlines), adaptability (treating weather conditions), trust and security (managing airports).

This section presents the related researches concerning the concepts of A-CDM, matching markets and algorithms of Game Theory, and optimization models for slot allocation in airport.

### 2.1 Collaborative Decision Making

In the 1990s, the philosophy of Collaborative Decision Making (CDM) was considered a new paradigm for the Air Traffic Flow Management (ATFM). It was designed based on the premise that an evolution in the processes of communication and information exchange between Air Traffic Control (ATC) agency and airlines would lead to better decisions in managing aircraft traffic [4]. At the time, the information exchange between Federal Aviation Administration (FAA) and airlines, both participants in the CDM, allowed the formulation of the current processes of Ground Delay Programs (GDP).

Usually, the scheduled flight operations are previously allocated to a takeoff/landing queue, comprising ATC slots. An ATC slot can be seen as a minimum amount of time required for an aircraft to be allowed to perform a takeoff or landing operation on the runway of a controlled airport [14].

The maximum number of aircraft that can land at an airport in a given period is known as the Airport Arrival Rate (AAR). The same analogy can be made in defining the rate of takeoff in an airport as Airport Departure Rate (ADR).

If it is detected that a sector of airspace will be congested at a certain time of the day, the traffic controller must apply appropriate measures, trying to reduce the number of aircraft at the affected location. This reduction is intended to maintain a safe amount of flights operating in the same controlled sector, avoiding congestion.

Although there are various restrictive measures such as ground holding delay, airborne holding delay, miles-in-trail, reroute, slot swapping, among others. For security reasons, preference is given to actions that involve solutions regarding ground holding. It is common sense the assumption that it is safer to change the conditions of flight of an aircraft that is in the ground than in the air [6, 13, 27].

When a ground delay program is applied, the AAR of some airports is reduced. Therefore, the incoming flights that should arrive during the scheduled times of congestion are delayed in their takeoffs. This restrictive measure brings a need of a change in the original slot allocation schedule of flights that will use the runways in these airports.

In this context, the GDP can be understood as a multi-stage process that deals with the management of slots queue allocation in airports impacted by operational capacity constraints. It is based on algorithms and information exchange between agents, being defined and applied by ATC agency with the participation of airlines [27].

### 2.2 Matching Markets

Game Theory has been used as a mathematical theory for modeling and analysis of the strategies among multiple players by economists, mathematicians, biologists and computer scientists and others to develop the applications with considerable social contribution [16, 26]. In recent years, Game Theory has become the focus of several researches in transportation studies [3, 4, 19, 23].

One of the reasons for the success of this theory is due to the diversity of theoretical and real scenarios that it can be applied. For example, we can mention the study of stock market, the dominance of genes in genetic evolution, regional war conflict, election results, economic markets, among others [4, 16, 25].

In economics, it is used to study the relationship between supply and demand of resources in societies. However, some researchers use it to analyze the behavior of allocation algorithms, enabling the distribution of these resources among agents in specific settings [22].

Since 1950's, Game Theory has been used to solve a wide range of problems, such as hiring processes in the labor market, students' admissions in the universities, network and internet design, organ allocation among patients and donors, among others [10,11, 22].

As the use of runways of an airport can be considered as a limited resource of aeronautical and airport infrastructure, the matching markets models can be associated to ATFM processes considering the demand and capacity of the runway for aircraft. Therefore, the allocation of slots, both for landing or takeoff operations can be modeled as a "market".

Although this association seems intuitive, few studies have so far been presented in ATFM. It is a challenge to exploit the potential of the matching approach for enhancement of Airport Collaborative Decision Making (A-CDM).

### 2.3 Optimization Models for Slot Allocation

The solution of delaying aircraft at the airport to deal with capacity issues is a complex problem known as Ground Holding Problem (GHP), in which the aircraft will be affected and the delay time assigned to each aircraft.

Although there are several methods proposing numerous solutions, ranging from operation research to multi-agent systems, the Ball et al. [4] and Wolfe et al. [29] researches indicate that there is, for a while now, a trend of using optimization models based on Game Theory to attend the evolution of the A-CDM.

Rassenti et al. [18] developed a combinatorial auction mechanism for airport slots; Ball et al. [4] resumed the study, analysis of objectives and concerns regarding aviation auction problems. And Balakrishnan [3] developed two solutions based on market models using Top Trading Cycle (TTC) and Vickrey-Clarke-Groves (VCG) pricing mechanisms.

These innovative studies showed a significant contrast between the proposal and the techniques that are currently using in ATFM. A process using monetary transfers between airlines, during slot allocation, is considered a significant change in the current paradigm. In order to determine the acceptability of these models, a more detailed analysis of exchange policy is required, as well as taking traffic regulators and other participants in the CDM to have a new perspective of the process.

In their recent work, Cruciol et al. [8] developed reward functions to evaluate the performance related to aircraft on ground and in the air management, ground delay control and complexity analysis of air sectors. Ribeiro and Weigang [19] presented a solution based on Game Theory to management the takeoff sequence of aircraft at airports. The proposed decision support system called Collaborative Departure Management (CoDMAN), was developed under the CDM philosophy.

On the other hand, the matching approach has been applied to ground delay problems which was conceptualized based only on the Top Trading Cycle (TTC) algorithm [24]. This mechanism considering one-sided markets, has been defined as aircraft of airlines oriented models [3, 23].

In the model proposed by Balikrishnan [3], the players were defined as agents by individual aircraft. The author has narrowed the solution to meet specific objectives for each flight, without taking into account the strategic decisions of each airline. More recently, the model proposed by Schummer and Vohra [23] to define the agents as airlines, dealing with slots relocation between their aircraft and expanding the CDM's "ownership" concept. Considering the innovation research, the study discussed one-sided market, in which only the interests of ATC agency and airlines are predicted in its architecture.

## 3. CDM ALGORITHMS

The ground delay program (GDP) is a process carried out in three steps. Two of these steps are executed by algorithms implemented with different functions, as shown in Figure 1. This process is implemented in partnership between the Federal Aviation Administration (FAA) and airlines according to CDM, where the interests of the parties are fulfilled [6].

In the classic CDM model, ATC agency and airlines are main partners in the collaborative decision making. In this process, the airlines provide trusted, reliable, and up-to-date information to traffic controllers, such that a better outcome of slot allocation can be achieved.

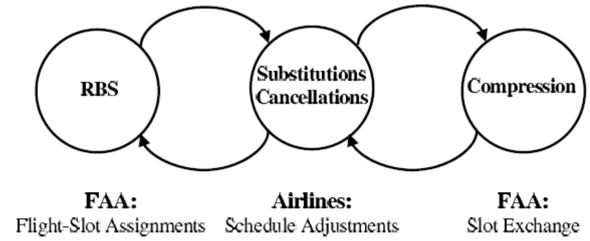



Figure 1: Classic CDM architecture [27].

After the reduction of airport arrival rate (ARR) by a preset time, representing the capacity of the runway configuration in the affected airport, the number of aircraft that will operate at that location is also reduced. Therefore, the first step of classic CDM involving GDP implements the redistribution of slots among the new number of aircraft that can operate per hour at the airport.

The Ration-By-Schedule (RBS) algorithm in Classic CDM intends to create a new schedule for the allocation of slots with revised times, and allocates the flights originally presented based on the new schedule. This allocation preserves the original order of arrival flights and that is defined for each aircraft [27]. Figure 2 shows an example of the application of the RBS algorithm.

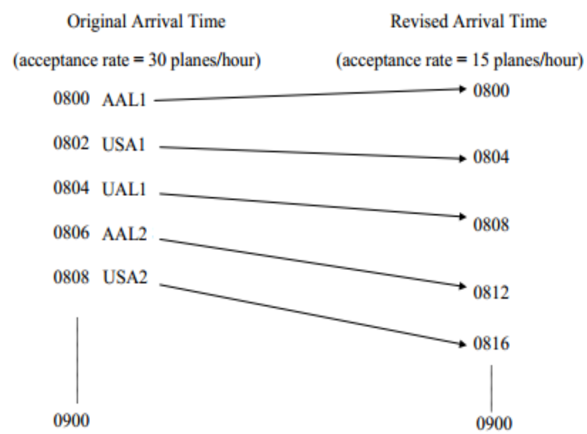



Figure 2: The application of RBS algorithm [6].

It is important to note that the effect of delays on aircraft is cumulative. For example, if the AAR capacity at any airport is reduced from 30 to 15 flights per hour, it will not only result in a 4 minute delay for each aircraft affected by GDP. The first aircraft in the new schedule will not suffer any delay, the second aircraft will use the runway 2 minutes later than in the original schedule, while the third aircraft will operate with a delay of 4 minutes, and so on. Therefore, in the new schedule, the tenth and twentieth aircraft will be assigned a delay of 36 and 76 minutes, respectively, compared to their arrival times as originally planned.

The opportunity thus for airlines to enables the analysis of the results reported by the algorithm, and making strategic decisions in order to mitigate the adverse effects of a GDP on their flight operations. Therefore, in the step named *Substitutions and Cancellations*, it is the airlines' responsibility to communicate on time: a) the possible delays due to mechanical failures and other operational problems, b) the cancellations due to internal adjustments and strategic decisions by airlines on their flights, and c) the replacements of flights among slots "owned" by the same airline, in which a flight can be prioritized over another.

After this second step, the new schedule created by the restrictions imposed by the GDP may contain "holes" due to the cancellations, which might in turn leave slots with no flights assigned to them.

To optimize the process, in which some slots from the current schedule would not be used, an algorithm was created, known as *Compression*, which fills the gaps in vacant slots according to pre-defined rules between ATC agency and the airlines.

The *Compression* algorithm works as follows: when a slot is vacated, the Compression tries to allocate it with another flight from the same airline that "owns" that slot. If the algorithm finds a feasible flight, it performs the exchange, but if there are no flights available, then the algorithm will seek a flight that belongs to another airline. If such a flight is found, the algorithm will allocate it in the slot, also changing the slot "ownership" between the airlines. If no flight is found, the algorithm will simply declare the slot as unused.

The algorithm must handle restrictive slot swapping parameters for a flight to be considered "feasible", such as minimum operating hours and minimum times for arrival at airports [13].

According to [6], this model was built through some basic concepts in CDM philosophy: the concept of "property", by which each airline has total control over its slots, without invading the allocations of competing companies, the concept of "priority", by which flights of the airline that owns the vacant slot are handled first, and the concept of " justice", by which each airline receives a percentage of slots equal to the percentage that it had in the original flight schedule (Official Airline Guide - OAG). More information on this process can be found in [6, 13, 27].

Since its adoption, the Compression algorithm has presented several limitations on the CDM philosophy, regarding its use. According to Schummer and Vohra [23], the algorithm does not guarantee that airlines report, in some cases, their flight cancellations. This situation causes the slots to become unusable, since the competing airlines cannot reallocate their flights to better positions in the schedule. In more serious cases, the algorithm can generate unstable results.

## 4. DA-CDM MODEL

The model proposed in this paper is based on the allocation mechanism proposed by Gale and Shapley [12], which became known as Deferred Acceptance. The choice of this mechanism is justified by its maturity in solving practical problems in two-sided matching markets, dating from the 1950s, and by its wide usage in complex scenarios that includes from geographical restrictions to compatibility between organ donors [20, 21, 22].

Since the ground delay program (GDP) can be seen as a problem of resource allocation, the environment can be characterized as a "slots market" in which there are two sets, one representing a group of flights and another a group of slots. Dealing with this market by using the DA-CDM model aims to assign each flight to each slot, through a "one-to-one" relationship, respecting the preferences of each allocation, which leads to a stable result.

### 4.1 Agents Selection

Decision makers in this model have been defined based on key players from “slots market" and studies undertaken by Norin [17] on the ATM stakeholders. In his work, the importance of ATC agencies and airlines as active participants in the CDM philosophy becomes explicit, together with the airport manager in the airport infrastructure, impacted by a ground delay program (GDP). These agents are defined as:

- ATC Agent: is characterized by a single agent responsible for detecting congestion in advance by predicting aircraft occupancy in the air scenario using data available in the flight schedule. Its goal is to control and optimize the traffic flow, applying the security measures at airports when necessary by ATC agency.
- Airline Agents: are agents that have flights that will be operating in a given day. Each agent’s goal is to control its aircraft with regards to planned times of takeoff and landing, reporting possible schedule changes due to technical and/or mechanics operational problems, or cancellations that may interfere in the original flight schedule.
- Airport Agents: are agents represented by the airports of origin and destination, defined in the flight schedule. Their goal is to maintain the appropriate flow of takeoffs and landings in their runways, adapting to the operational capacity restrictions specified by the ATC Agent.

It is important to note that the ATC agent represents a centralizing agent in the market and has no preferences for allocation over any elements in the scenario, due to safety and aircraft traffic flow concerns. In this matter, the Airline and Airport agents can be characterized as decision-makers in the slots allocation problem. They are responsible for determining strategies based on their own goals, in order to enable the correct formulation of the new schedule for airport runways use, in a moment of GDP.

### 4.2 Reward Structure

Each agent group’s goal in this market may be different and even contradictory. For airlines, it’s important to reduce the total delay of their flights, reducing the costs inherent to these delays, prioritize strategic flights over others, treating differently passengers in international flights or with stopovers, etc. As for the airport's concessionaire, maybe the goal is to optimize the aircraft flow in the apron, to enlarge the rate of passengers’ arrivals, to prioritize flights already en route, among others.

As an initial proposal, a simple approach was defined in order to model the objective function of Airlines agents, according to Equation 1. In this definition, a strategy focuses on the operating profit of each aircraft belonging to a set of flights of a given airline.

$$R_F(f) = \alpha(f)\left[\left(\sum_{k=1}^{q} sr(p_k) - vc(p_k)\right) - fc(f)\right] \qquad (1)$$

where *sr* is the sales revenue, *vc* is the variable cost and *fc* is the fixed cost per passenger *p* of flight *f*, for a total of *q* passengers of the same flight. The function α is the importance given to flight *f* by its airline, with a value x, where 0 < x <= 1. This function allows the airlines with the possibility to prioritize some flights over others. Thus, policies have not addressed by Equation 1, because it is still not knew how the destination of the flight, the aircraft size, etc. can be treated. In this configuration, the higher the $R_F$ value, the more profitable is the flight for the airline responsible.

The objective function of the Airport agent is shown in Equation 2 based on a strategy that prioritizes flights according to the amount of passengers and to the aircraft’s delay time. This policy allows the decongestion from inside the airports of origin and the improvement of people flow expected in the destination airport. Moreover, it helps to reduce the stress on crew and passengers of each flight.

$$R_S(f) = \beta(f)\; q^{\theta(t-at(f),\; c)} \qquad (2)$$

where *t* represents the current time, *at* is the estimated time of arrival, *q* is the total number of passengers of flight *f* and *c* is an adjustment constant. The $\beta$ function is the importance given by the airport manager to flight $f$, with a value x, where $0 < x <= 1$.

The θ function aims to process the result of the difference between the times $t$ and $at$, in minutes [2, 8]. If the calculation is zero or negative, indicating that the flight is not delayed, the θ function returns the value 1. If the calculation is positive, this value is divided by the adjustment constant $c$, and the function θ returns the integer portion of the value. For example, if a flight's estimated time of arrival is 09:30 and it's now 11:00, the 90 minutes difference will be divided by $c$. If $c$ is equal to 30, the function θ returns the value 3. If $c$ is 60, the θ function returns the integer part 1. Therefore, the higher the value of $c$, the lower is the importance given to the flight delay.

It is important to note that the equations presented allow us to set a priority to flights affected by ground delay program (GDP), enabling an ordering among them.

## 4.3 Formal Definition

A market of slots with one-to-one relationship is formed by $< F, S, \succ_F, \succ_S >$, where $F$ and $S$ are disjoint and finite sets of allocable elements, where $F$ represents flights $f_1, f_2, \ldots, f_m \varepsilon F$ and $S$ represents slots available in the market $s_1, s_2, \ldots, s_n \varepsilon S$, containing $m$ e $n$ elements separately.

The elements of the set of arrival slots $S = \{1, 2, 3, \ldots, |S|\}$ can be interpreted as ordinal representations of time: for $s, v \varepsilon S$, where $s < v$ means that a slot $s$ represents a time earlier than the slot $v$.

The earliest possible arrival time (EPAT) for flight $f \varepsilon F$ is denoted by $e_f \varepsilon S$. Therefore, the flight $f$ might be assigned to slot $s_i \varepsilon S$ where $i = 1, \ldots, |S|$, only if $e_f \leq s_i$. If the EPAT of a flight $f$ is 10:00am, it will never be able to land in a 09:30 slot at the destination airport.

Each flight $f_j$, where $j = 1, \ldots, |F|$, has a strict, complete, and transitive preference $\succ_F$ over the elements of the other set. The same analogy can be drawn about the preference lists of slots $\succ_S$.

By "complete", it can be understood that all the elements from a set can sort all the elements from the other set in relation to any possible choice, without presenting any indecision in the ordering. By "strict" preference, we mean that the elements of this market should be able to classify each element of the opposite set according to a strict preference order, i.e., without indicating indifference between them. As for "transitive", we understand that there is a consistency in the choices made based on the preferences in a set.

The individual lists containing the ordered preferences can be represented as a set $P(f)$ where $P(f) = s_2 \succ s_1 \succ s_3 \succ \ldots \succ s_n$ means that the flight $f$ strictly prefers to be allocated to slot $s_2$ rather than slot $s_1$. If the flight $f$ cannot be allocated to slot $s_2$ so it prefers to be allocated with $s_1$, and so on.

Allocation preferences are defined by *decision maker agents* using equations 1 and 2, in which airlines are responsible for $\succ_F$ preferences for each of their flights $f$ and the airport affected by GDP is responsible for the $\succ_S$ preferences of each slot $s$.

A matching is the result of this market, represented by the association of elements from a set with elements from another set through the function $\mu: F \cup S \rightarrow F \cup S$ such that $\mu(f) = s \Leftrightarrow \mu(s) = f$, for all $f \in F, s \in S$.

A "blocking pair" is formed by the pair $(f, s) \varepsilon F x S$ if both prefer each other rather than their pairs formed in the matching $\mu$, i.e., $s \succ_F \mu(f)$ e $f \succ_S \mu(s)$.

A matching is "stable" if it presents a satisfactory allocation for all elements of the sets, where there is no blocking pair.

## 4.4 DA-CDM Allocation Algorithm

Using Deferred Acceptance algorithm, the allocation algorithm is modeled to run after the substitutions and cancelations step. There are two procedures: pre-processing and main steps.

**Algorithm** Pre-processing

**Input:** sets $F$ and $S$, where:

- $F$ represents the set of flights $f$, such that $f \in F$, and;
- $S$ represents the set of slots $s$, such that $s \in S$.

**Output:** $< F, S, \succ_F, \succ_S >$, where:

- $\succ_F$ represents the set of preferences of each $f \in F$, and;
- $\succ_S$ represents the set of preferences of each $s \in S$.

1: Based on a list of flights $F$, the airport defines a list of $\succ_S$ preferences for each slot $s$, guided by strategic premises, according to equation 1;

2: Each airline defines the preferences order for their flights $f$ according to available slots in $s$. In this model, we use equation 2.

After the formulation of the necessary information for the slot reallocation processes, the main algorithm tries to achieve a result that provides a stable matching, considering the preferences of each element in the market.

**Algorithm** DA-CDM Allocation

**Input:** a slot market $< F, S, \succ_F, \succ_S >$, representing a new landing schedule, updated according to the RBS process and the informations regarding delays, cancelations and substitutions from the previous stages.

**Output:** a new schedule of stable landings $\mu$: $F \cup S$.

1: Each flight $f \in F$ makes an allocation offer to its preferred slot, according to:

a) The feasible arrival time rule, where $s \geq e_f$, and the order in its preference list $\succ_f$.

2: Each slot $s \in S$ accepts its preferred proposal, rejecting all the others, according to:

a) The feasible arrival time rule, where $s \geq e_f$, and the order in its preference list $\succ_s$.

$j$: Any flight $f \in F$ that is rejected in the step $j-1$ makes a new allocation offer to its next preferred slot $s \in S$ that have not rejected it yet, according to rule 1a. Each slot $s \in S$ remains allocated to its best offer so far, rejecting any other, respecting rule 2a, $j = 1, 2, \ldots, m$, $m = |F|$.

Stop: When there are not new proposals to be made:

a) rationalize vacant slots, placing aircraft in better positions, respecting the order already defined and the feasible arrival time rule, where $s \geq e_f$.

b) remaining vacant slots are distributed among the owner airlines, respecting the original order of the algorithm;

c) the algorithm terminates.

With the feasible arrival time rule present in 1a and 2a, the algorithm ensures the correct processing in case of inconsistency in preference lists of flights and slots, represented by the situation where $s < e_f$. It is important to note that the airlines and the airport are responsible for creating these rules, which are defined, in this paper, by Equations 1 and 2.

As $e_f$ is based on the original time of each flight, we have two different scenarios that the algorithm does not need to address: a) the situation where the feasible arrival time of a flight is the last slot in the schedule, because for all flights $f \varepsilon F$, each $e_f < |S|$; b) there is not a situation where two flights have the same arrival slot at the destination airport, because for all flights $f, f' \varepsilon F$, $e_f \neq e_{f'}$.

The algorithm must always follow the ordered preference lists of all allocable elements in the model. According to the "Stop" step, each flight is definitely allocated to the slot it was associated with in the last step of the algorithm, where the result is always a stable matching. The proof of stability and stopping for the allocation mechanism for two-sided matching markets is given by Gale and Shapley [12].

## 5. COMPARISON AND DISCUSSION

In this section we present the solutions to an analytic example of a ground delay program. The purpose hereof is illustrating the performance of our proposed methodology and to compare the different features of the Classic CDM and DA-CDM models.

After running the Ration-By-Schedule (RBS) algorithm and the Substitutions and Cancellations step (see Figure 1), performed by airlines, suppose the initial scenario of the third step is as shown in Table 1.

**Table 1. Initial setting for step 3 of the GDP.**

| SLOT | Flight | Airline | $e_f$ |
|---|---|---|---|
| $s_1$ | empty | A | |
| $s_2$ | empty | B | |
| $s_3$ | $f_3$ | C | 1 |
| $s_4$ | $f_4$ | B | 1 |
| $s_5$ | $f_5$ | A | 2 |
| $s_6$ | $f_6$ | D | 5 |

This example shows four aircraft belonging to the airlines A, B, C and D, respectively, competing for six slots of which two are vacant due to flight cancellation in the previous step, as well as Substitutions and Cancellations. The feasible arrival time of each aircraft is shown in column $e_f$. This schedule is based on the time originally scheduled for flight $f$, and represents the restriction that the flight can only get to slot $s$ in the destination airport if $s \geq e_f$.

Based on this information, in a preprocessing step, the algorithm defined in the DA-CDM model creates preference lists where, by Equation 1, airline agents are responsible for the aircraft and, according to the rules defined in Equation 2, the airport agent is responsible for slots preferences. For illustrative purposes only, we hypothetically define the preferences of all allocable elements (flights and slots) as shown in Table 2.

**Table 2. Preferências dos voos e SLOTS.**

| Airline Agent | Airport Agent |
|---|---|
| | $P(s_1) = \{ f_5 \succ f_4 \succ f_3 \succ f_6 \}$ |
| $P(f_3) = \{s_1 \succ s_3 \succ s_2 \succ s_4 \succ s_5 \succ s_6\}$ | $P(s_2) = \{ f_5 \succ f_3 \succ f_4 \succ f_6 \}$ |
| $P(f_4) = \{s_1 \succ s_3 \succ s_2 \succ s_4 \succ s_6 \succ s_5\}$ | $P(s_3) = \{ f_6 \succ f_4 \succ f_3 \succ f_5 \}$ |
| $P(f_5) = \{s_3 \succ s_6 \succ s_4 \succ s_1 \succ s_5 \succ s_2\}$ | $P(s_4) = \{ f_5 \succ f_6 \succ f_3 \succ f_4 \}$ |
| $P(f_6) = \{s_4 \succ s_2 \succ s_1 \succ s_3 \succ s_5 \succ s_6\}$ | $P(s_5) = \{ f_6 \succ f_5 \succ f_3 \succ f_4 \}$ |
| | $P(s_6) = \{ f_4 \succ f_5 \succ f_6 \succ f_3 \}$ |

At this point, the main processes from Classic CDM and DA-CDM perform as follows:

**Compression 1**: starts by searching for flights from airline $A$ that may be allocated to $s_1$. The only active flight from airline $A$ is $f_5$, but it cannot be allocated because its feasible arrival time ($e_f$) is $s_2$. Therefore, since $A$ has no more feasible flights, the flight from the next company that can be assigned to $s_1$ is $f_3$ from airline $C$. After performing the swapping, the algorithm also exchanges the slot's ownership between airlines.

**DA-CDM 1**: flights $f_3$ and $f_4$ make allocation proposals to slot $s_1$, and flight $f_5$ makes a proposal to $s_3$, which are, according to the preference lists in Table 2, all their first choices. Flight $s_6$ would like to propose an association with $s_4$, but since its $e_f$ is 5, its proposal is directed to $s_5$, in accordance with the algorithm's first rule. The slot $s_1$ accepts $f_4$'s proposal, which is its most preferred flight, rejecting flight $f_3$. The slot $s_3$ accepts $f_5$'s proposal, those being its only proposals so far, and $s_5$ accepts flight $f_6$ under rule 2a.

After the processes' execution, the resulting allocation from the first cycle is shown in Table 3.

**Table 3. Compression x DA Algorithm (end of cycle 1).**

| | CDM | | | DA-CDM | | |
|---|---|---|---|---|---|---|
| ***SLOT*** | ***Flight*** | ***Airline*** | $e_f$ | ***Flight*** | ***Airline*** | $e_f$ |
| $s_1$ | $f_3$ | C | 1 | $f_4$ | B | 1 |
| $s_2$ | empty | B | | | | |
| $s_3$ | empty | A | | $f_5$ | A | 2 |
| $s_4$ | $f_4$ | B | 1 | | | |
| $s_5$ | $f_5$ | A | 2 | $f_6$ | D | 5 |
| $s_6$ | $f_6$ | D | 5 | | | |

Under this scenario, the processes run again as following step 2:

**Compression 2**: in this moment, the algorithm verifies that the slot $s_2$, which is vacant, belongs to airline $B$ and its flight $f_4$ can be moved to $s_2$, according to his feasible arrival time. Therefore, the algorithm executes the swapping.

**Table 4. Compression x DA Algorithm (end of cycle 2).**

| | CDM | | | DA-CDM | | |
|---|---|---|---|---|---|---|
| ***SLOT*** | ***Flight*** | ***Airline*** | $e_f$ | ***Flight*** | ***Airline*** | $e_f$ |
| $s_1$ | $f_3$ | C | 1 | $f_4$ | B | 1 |
| $s_2$ | $f_4$ | B | 1 | | | |
| $s_3$ | empty | A | | $f_3$ | C | 1 |
| $s_4$ | empty | B | | | | |
| $s_5$ | $f_5$ | A | 2 | $f_6$ | D | 5 |
| $s_6$ | $f_6$ | D | 5 | | | |

**DA-CDM 2**: in this new cycle, the flight $f_3$ makes an allocation proposal to slot $s_3$, second in the preference list and not yet rejected. Even though $s_3$ is assigned to $f_5$, according to its preference list, the slot $s_3$ prefers to be allocated to $f_3$ rather than to $f_5$. Thus, it dispenses with the flight $f_5$ and gets $f_3$. Table 4 shows the result of the end of cycle 2.

Now it is possible to verify the execution of the rest processes.

**Compression k**: the next vacant slot belongs to airline $A$ and among its flights that can be allocated to it. The flight $f_5$ is chosen by the algorithm due to $e_f \leq s_3$. The slot $s_4$ remains vacant because there are no flights that could be allocated and flight $f_6$ is allocated to $s_5$, respecting its feasible arrival time of 5.

**DA-CDM k**: as the flight $f_5$ was rejected in the previous cycle, now it makes a proposal to the next slot on its preference list that has not yet rejected it. Therefore, $f_5$ is allocated to $s_6$. As there are no more proposals to be made, by the "Stop $a$" step, $f_3$ can be moved to $s_2$, and $f_5$ can be moved to $s_3$. Since the $e_f$ from flight $f_6$ is 5, its position cannot be improved. Meanwhile, by the “Stop *b*” step, the vacant slots are distributed among its owner airlines, according to their original order in Table 1.

As the vacant slots cannot be allocated to any more active flight, the algorithms terminate with the schedule presented in Table 5.

**Table 5. Compression x DA Algorithm (end of the process).**

| *SLOT* | CDM | | | DA-CDM | | |
|---|---|---|---|---|---|---|
| | *Flight* | *Airline* | $e_f$ | *Flight* | *Airline* | $e_f$ |
| $s_1$ | $f_3$ | C | 1 | $f_4$ | B | 1 |
| $s_2$ | $f_4$ | B | 1 | $f_3$ | C | 1 |
| $s_3$ | $f_5$ | A | 2 | $f_5$ | A | 2 |
| $s_4$ | empty | B | | empty | A | |
| $s_5$ | $f_6$ | D | 5 | $f_6$ | D | 5 |
| $s_6$ | empty | A | | empty | B | |

This hypothetical scenario enables the monitoring of both algorithms operation. It is important to note that, in the DA-CDM model, airlines *B* and *C* were not rewarded or punished in the allocation DA-CDM process. All allocations were made respecting both preferences of airlines on flights, and of the airport on slots. The original order of vacant slots was also remained by the end of the process, enabling a more equitable allocation for airlines. This is important in situations where the algorithm needs to be reprocessed due to dynamic changes in the air scenario.

Based on this example, on the execution of each process, and on the evidence from literature (see Section 3 and 4), the main features of both models can be verified as shown in Table 6.

**Table 6. Comparison between Classic CDM and DA-CDM.**

| Items | Classic CDM | DA-CDM |
|---|---|---|
| *Agent ATC* | Deals with runway use restrictions, imposed on airports. | To deal with runway use restrictions, imposed on airports. |
| *Agent Airline* | Do not have strategic preferences over aircraft allocation. | With the strategic preferences over aircraft allocation. |
| *Agent Airport* | Not mentioned. | With the strategic preferences over slots allocation. |
| *Arrival slots* | Are filled whenever possible. | To be filled whenever possible. |
| *Property* | If an airline cannot use its available slot, it is always compensated with slot “ownership” to exchange with another airline that owns a flight available. | The airlines retain ownership over their vacant slots at the end of the process, ensuring the original order of slots. |
| *Priority* | The flights from the airline that owns the vacant slot are considered before the flights of other airlines. | All flights have the same priority in the process. |
| *Justice* | At the end of the process, each airline has the same percentage of slots they did at the beginning of the process. | At the end of the process, each airline has the same percentage of slots what they did at the beginning of the process. |
| *Slots loss* | There is no possible way an airline loses involuntarily a slot that it owns. | There is no possible way for an airline to lose involuntarily a slot that it owns. |
| *Order of operation* | The order by which flights are chosen to operate impacts on the final result of allocation. | The order by which flights are chosen to operate does not impact on the final result of allocation. |
| *Estability* | It may produce unstable results. | It always finds a stable result. |

As showed in Table 6, both methods have positive and negative aspects. The proposed model solves the slot allocation problem using Game Theory. The algorithm developed in this paper allows one more ATM stakeholder participate of the GDP process enhancing classic CDM concepts. It is important to mention that using Game Theory all agent preferences are respected by the new algorithm.

## 6. CONCLUSIONS AND FUTURE WORK

We present a Deferred Acceptance CDM model using a matching approach for airport collaborative decision making (A-CDM) with the participation of three agents: ATC agency, airlines and airport managers. As the ground delay program (GDP) is a sophisticated process with dynamic online control property and limited slot resources, the mechanism of two-sided matching markets demonstrates a suitable solution to allocate flight slots in Airport CDM. The proposed model also involves a new player such as the airport managers concerning the restrictive measures in the application of ground delay program.

Comparing to the Compression algorithm in classic CDM, the DA-CDM algorithm aims to assign each flight to each slot, through a "one-to-one" relationship, respecting the preferences of each allocation. This leads to a stable allocation in the case of flight delay(s), as well as in other cases. The main benefits for the partners in CDM and the advantages of the developed model can be summarized as:

- For the ATC agency, the DA-CDM model provides the allocation results by a reliable process including ground delay program (GDP), in which the standards of flow and flight safety are maintained.
- For airlines, the DA-CDM model provides the allocation results for aircrafts by an efficient process directly to reduce the operation cost in taxiing, fuel, crew expenses, and also to reduce the impact to environment.
- For airport managers, the DA-CDM model involves their participation in the decision making process to help the management and optimization of airport resources by improving the fluency of aircrafts on runways, coordination on the apron and the passengers´ movement through gates, among others.
- Even the DA-CDM model does not involve the decision participation of the passengers, the application of the developed model can reduce the delays by applying coordinated actions between airlines and airports achieving a greater proximity of the flights’ original departure and arrival times.

Besides allowing the participation of key agents in the ground delay program, DA-CDM model also allows the definition of preferences of airlines and airport managers to allocate a aircraft to a slot that are respected by the Deferred Acceptance algorithm. This is an important feature of the matching mechanism that can be used to create the possibility of defining specific roles for each agent. This advantage is for ATC agency, airlines, and airport managers to develop the local strategies in a global solution.

It is important to note that, in the current A-CDM application, the airport managers are absent in the decision process. In our proposed model, airport managers are included as a decision agent in the A-CDM process. They affect and are affected by GDP involving the processes of takeoff and landing. The airport managers are also responsible for ground handling of aircraft and services for passengers, such as airport operators, aircraft operators, and ground service handlers, among others. This part should be also included in the A-CDM process.

As the future work, the DA-CDM may be modified with the capacity to define the preferences from airport managers such as approach managers (APP), tower, ground, and other managers from various airport services. Further, the analysis of time and complexity on the algorithm could be provided and DA-CDM should be modified with the capacity to get the optimization results via Pareto efficiency. In the application of the DA-CDM, some performed tests should be also considered with the different purposes for each agent. For example, the allocation effects on aircraft can be analyzed in difference scenarios. Attention could also be given to handling the possible coalition between airlines by using real data from the Brazilian Air Navigation Management Center (CGNA).

## 7. REFERENCES


[1] Agogino A. and K. Tumer. Regulating Air Traffic Flow with Coupled Agents Advances in Complex Systems, In *Proc. of 7th Int. Conference on Autonomous Agents and Multiagent Systems*, 535-542, Estoril, Portugal, 2008.

[2] Arruda Junior, A. C., Weigang, L. and Barros, A. Fairness analysis with flight cost impact using reinforcement learning approach. *Journal of the Brazilian Air Transportation Research Society*, 8, 9-27, 2012.

[3] Balakrishnan, H. Techniques for Reallocating Airport Resources during Adverse Weather. In *Proc. of the IEEE Conference on Decision and Control*, 2949 - 2956, New Orleans, USA, 2007.

[4] Ball, M., Donohue, G., and Hoffman, K. *Auctions for the safe, efficient and equitable allocation of airspace system resources*. In Cramton, P., Y. Shoham and R. Steinberg, eds. *Combinatorial Auctions*, MIT Press, Cambridge, pp 507-538, 2005.

[5] Ball, M. O., Hoffman, R. L. and Mukherjee, A. Ground Delay Program Planning Under Uncertainty Based on the Ration-by-Distance Principle. *Transportation Science*, 44(1):1-14, 2010.

[6] Butler, T. D. *Optimization Model with Fairness Objective for Air Traffic Management.* NEXTOR report, University of Maryland, College Park, MD, 112 p, 1998.

[7] Crespo, A. M. F., Weigang, L., Barros, A. Reinforcement learning agents to tactical air traffic flow management. *International Journal of Aviation Management*, 1(3), 145-161, 2012.

[8] Cruciol, L. B. V. , Arruda Junior, A. C., Weigang, L., Li, L., Crespo, A. F. Reward functions for learning to control in air traffic flow management. *Transportation Research. Part C, Emerging Technologies*, 35, 141-155, 2013.

[9] Dib, M. V. P., L. Weigang, L., A. C. M. A. Melo. 2007, Approach of Balancing of the Negotiation among Agents in Traffic Synchronization. *IEEE Latin America Transactions*, 5, 338-345, 2007.

[10] Ergin, H. and Sönmez, T. Games of School Choice under the Boston Mechanism. Mimeo, MIT. *Journal of Public Economics*, 90, 1-2 (January), 215-237, 2006.

[11] Gai, T., Lebedev, D., Mathieu, F., Montgolfier, F., Reynier, J., and Viennot, L. Acyclic preference systems in P2P networks. In *13th European Conference on Parallel and Distributed Computing*, LNCS 4641, 825-834, 2007.

[12] Gale, D. and Shapley, L. College Admissions and the Stability of Marriage. *American Mathematical Monthly*, January, 69(1), 9-15, 1962.

[13] Hoffman, R. L. *Integer Programming Models for Ground-Holding in Air Traffic Flow Management.* Dissertation, PhD, Department ISR, NEXTOR, 1997.

[14] ICAO. DOC 9854: *Global Air Traffic Management Operational Concept. International Civil Aviation Organization.* DOC 9854-AN/458, 2005.

[15] INFRAERO. *Concessão de Aeroportos*. http://www.infraero.gov.br/index.php/transparencia/concessao.html, 2013.

[16] NOBEL. *Stable matching: Theory, evidence, and practical design.* The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2012, Alvin E. Roth, Lloyd S. Shapley, Advanced Information, 2012.

[17] Norin, A. *Airport Logistics: Modeling and Optimizing the Turn-Around Process*. Licentiate thesis, monograph, Department of Science and Technology, Linköping University, SE-601 74 Norrköping, Sweden, 2008.

[18] Rassenti, S., Smith, V., and Bulfin, R. *A combinatorial auction mechanism for airport time slot allocation.* Bell Journal of Economics, 13(2), 402-417, 1982.

[19] Ribeiro, V. F. and Weigang, L. Collaborative Decision Making with game theory for slot allocation and departure sequencing in airports. In *17th Air Transport Research Society World Conference*, Bergamo, Italy, 2013.

[20] Roth, A. E. and Peranson, E. The Redesign of the Matching Market for American Physicians: Some Engineering Aspects of Economic Design. *American Economic Review*, 89(4): 748-780, 1999.

[21] Roth, A. and Sotomayor, M. 1990. Two-sided matching: A study in game-theoretic modeling and analysis. *Econometric Society Monograph* 18. Cambridge, Cambridge University Press.

[22] Roth, A. E., Sönmez, T. and Ünver, M. U. Kidney Exchange. *Quarterly Journal of Economics*, 119(2), pp. 457-488, 2004.

[23] Schummer, J. and R. V. Vohra. Assignment of Arrival SLOTS. *American Economic Journal: Microeconomics*, 5(2): 164-85, 2013.

[24] Shapley, L. and Scarf, H. On Cores and Indivisibility. *Journal of Mathematical Economics*, 1, 23-28, 1974.

[25] Sönmez, T. and Ünver, M. U. *Matching, Allocation, and Exchange of Discrete Resources*. Handbook of Social Economics, Vol. 1A. The Netherlands: North-Holland, 781-852 edited by Jess Benhabib, Alberto Bisin, and Matthew Jackson, 2011.

[26] Von Neumann, J. and Morgenstern, O. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[27] Vossen, T. and Ball, M. *Optimization and mediated bartering models for ground delay programs.* Naval Research Logistics, 53(1):75-90, 2006.

[28] Weigang, L., Dib, M. V. P., Alves, D. P., Crespo, A. F. Intelligent computing methods in Air Traffic Flow Management. *Transportation Research. Part C, Emerging Technologies*, 18, 781-793, 2010.

[29] Wolfe, S. R., Jarvis, P. A., Enomoto, F. Y., Sierhuis, M., Putten, B., and Sheth, K. S. *A Multi-Agent Simulation of Collaborative Air Traffic Flow Management.* Multi-Agent Systems for Traffic and Transportation Engineering, p. 357-381, edited by Ana L. C. Bazzan and Franziska Klügl, Information Science Reference, 2009.

# On the benefit of collective norms for autonomous vehicles

Vincent Baines, Julian Padget
Dept. of Computer Science,
University of Bath
{v.f.baines,j.a.padget}@bath.ac.uk

## ABSTRACT

This paper considers the (intelligent) vehicle domain from the perspective of situational awareness, but based on knowledge, rather than data, attempting to model the context of the (human) driver rather than that of an auto-pilot. We set out a (driver) simulation framework, in which some vehicles are operated by a collection of norm-aware BDI agents and connect this with the SUMO traffic simulation environment, which provides the background traffic. While the driver collective retains autonomy with respect to road conditions and actions, it receives guidance from several institutional models that implement social reasoning about the context in which the vehicle is currently situated. We demonstrate the benefit of rapid visualization of simulation metrics and use a range of domain-relevant metrics to show how it is possible to assess both collective (e.g. traffic flow) and individual impact (fuel consumption) arising from individual vs. institutional decision making.

## Keywords

multiagent systems, intelligent transportation systems, autonomous vehicles

## 1. INTRODUCTION

The ability of autonomous agents to operate in pursuit of both their own goals, as well as comply with obligations from a collective view, presents numerous challenges but a significant number of potential benefits. In order to explore what may be possible, a simulation framework has been established with a number of vehicle specific scenarios to assess both the suitability of the framework for such investigations, and to capture individual and global measurements of the effect of institutional governance in these scenarios.

An underlying assumption in the various scenario themes is that of knowledge exchange, both for the derivation of understanding about the environment, and the approach to how this data is shared between distributed components. The concept of Situational Awareness is adopted as a means to categorise information 'levels', considering Endsley's [14] concepts of perception, comprehension and projection as a transition from 'low' level information (e.g. a geographic xy location of another vehicle) to 'high' level information (e.g. given current speed and heading, there may be a collision based on the other vehicle's xy). We explore this theme and related concepts in Section 2.

The mechanism used to exchange these various information levels also needs consideration. A publish-subscribe mechanism has been adopted based on the Extensible Messaging and Presence Protocol (XMPP) [31] framework. Within this, information is packaged according to the Resource Description Framework (RDF), to add semantic annotation to the information exchanged, or JSON, where semantic information is not required. Coupled with a XMPP messaging server, this represents the nucleus of the simulation environment and is referred to as the Bath Sensor Framework (BSF). Supplemental tools have been built around this in order to assess data flow, from low level metrics (e.g. messages per second) through to a 3D representation of the environment and inferred 'high level' knowledge (e.g. collision volumes, upcoming traffic lights). More details about this aspect appear in Section 3.

The Belief-Desire-Intention (BDI) [10] model is adopted as the agent architecture in this work and specifically the Jason [9] platform, providing a multiagent system where agents store beliefs and available plans in order to pursue goals. In the context of the BSF framework, Jason is extended to process RDF data and pass it on to agents, who react accordingly, and can trigger actions back to the environment through creation of suitable RDF requests. The BDI model has been demonstrated in vehicle convoy scenarios (e.g. [27] and [3]) and that work is built upon further here.

In order to augment the capability of these agents to operate collectively and to be able to function in new situations about which they have no prior knowledge, the use of an institutional framework has been integrated into the BSF simulation. As an agent senses its world view via received RDF data triples, so does (each instance of) an institution, and whereas an agent may not have a suitable plan or belief handling for a given situation (e.g. socially complex or ambiguous cases) an institution, embodying situation-specific knowledge, can issue appropriate obligations to participants in order to achieve common goals. Furthermore, the institution is able to act as a situational governance mechanism, issuing obligations to individuals which might be contrary to the maximum satisfaction of their current desires, but of benefit to the wider collective (e.g. one vehicle being told to move out of the way to allow a queue to pass). We discuss the institutional aspect in more detail in Section 4

The opportunity to integrate such technology with real world vehicles increases as autonomous vehicles step ever closer to the mainstream. With Google's driver less car [24] and the Volkswagen based 'MadeInGermany' [16] vehicles both gaining mileage over the last few years, as well as more recent announcements such as Nissan's [17] there are autonomous vehicles across America, Europe, and Japan. Adopting vehicle scenarios as the chosen context provides an appropriate challenge for the simulation framework (i.e. high message rates and timeliness of message delivery) as well as a rich information context (i.e. higher level knowledge vs low level sensor feeds) with which to assess the application of both BDI agents and institutional frameworks.

Following the construction of a suitable simulation framework, and with the autonomous vehicle context in mind, two scenarios are put forward in Section 5 to explore the use of norms in the vehicle domain. The first investigates the use of an institution to transform a visual cue of a vehicle behind flashing its lights (requesting that the vehicle ahead moves to another motorway lane) to an obligation to change lane. The second explores the use of upcoming traffic light data based on a vehicle's current route and speed, and the use of an obligation to adjust speed in order to arrive at that light whilst it is green. Clearly, both such behaviours could be pre-loaded into the agent: the institution appears superfluous; our point is that such an argument can be made for *every* such scenario, which would lead to agents carrying a lot of plan baggage which may be rarely used and which, being embedded in the agent, is not readily revisable, furthermore, there will also always be scenarios not foreseen when the agent was constructed. Our position therefore, and what this paper seeks to demonstrate, is that through the delivery of obligations, institutions provide a mechanism for outsourcing agent knowledge of conventional and regulated situations, while permitting ready update and the provision of new knowledge on an as-needed basis [21]. Subsequently, we analyse some of the metrics collected from these scenarios in Section 6, which indicate a positive impact on fuel consumption. There is also some early indication that traffic flow can be improved, however further work is needed to establish and quantify this benefit using more realistic and demanding scenarios, as we outline in Section 7.

## 2. RESEARCH BACKGROUND

Whilst this work draws on a number of different research areas, the core theme is that of Situational Awareness. Formally, Endsley [14] defines this as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future" and this forms the basis of three levels of SA: perception, comprehension and projection. These levels are drawn on as knowledge representation levels within the framework and experimentation of work presented here. 'Low level' information is considered as the perception level (e.g. a traffic light x-y location), and as reasoning and data fusion is performed the information rises through the levels, firstly comprehension (e.g. distance to that traffic light from current position), through to projection (e.g. affect that light will have on vehicle given current speed and state of light).

With vehicles containing increasing technology in terms of driver aids and safety systems work has also been taking place to consider how cooperation between vehicles, based on V2V communication, could be beneficial. Coordination in terms of vehicle platooning or convoy behaviour has been receiving attention. The SAfe Road TRains for the Environment (SARTRE) study [6] demonstrated the ability of vehicles to form an effective convoy when following a designated lead vehicle, identifying benefits (e.g. time, fuel) and considering the real world implications of such message exchange. Given the physical limits encountered when using real networks in V2V communication [7] this provides motivation to explore whether we can communicate less via exchange of higher level information, and still provide acceptable knowledge transfer and performance. Such an approach is explored in the second scenario presented in this paper, which relates to the 'projection' aspect of SA based on traffic light state to future vehicle state. Particularly relevant to the first scenario put forward in this paper, Bilstrup [8] considers emergency vehicle routing, where V2V messaging is used to coordinate clearing a path for emergency vehicles.

Regarding vehicle coordination in relation to traffic lights, work has been undertaken [18] to implement communication between traffic lights and vehicles, in order to improve fuel consumption and reduce emissions. Similarly, a recent news announcement [30] provided details of Audi vehicles retrofitted with new technology interacting with traffic lights, in order to improve traffic flow. CO2 emission reductions of up to 15 percent are claimed, along with a potential 900 million litre fuel saving per year if the system were implemented throughout Germany, but no precise details of the simulation or the methodology are given, so it is not clear how the figures might be verified.

The use of institutions as a mechanism to provide norms in the absence of a clear individual choice, or as an enforcement mechanism contrary to the individual's choice, has been explored in contexts where an individual gains at the expense of peers [4], a scenario which can be easily applied to the vehicle domain. Furthermore, the possibility for multiple institutions to interact [12] (e.g obeying a traffic light vs. moving out of the way of an emergency vehicle) characterises scenarios where human drivers may struggle to resolve the situation. Indeed, the topic of human drivers interacting with autonomous vehicles will create even more challenges, and whilst thought has gone into what such hybrid interactions may look like (e.g. traffic light systems [13]) we believe there is a role for institutions in facilitating this integration as externally verifiable repositories of normative (conventional and regulatory) knowledge.

Considering specifically traffic situations, we have identified several future scenarios where the use of institutions could be of benefit. One such use could be to enforce variable speed limits, a technique currently implemented through the use of road signs with speed cameras as the enforcement mechanism. The benefits of such approaches have been assessed, for example on the M42 [25] and M25 [29] motorways in the UK, with findings [29] that whilst some objectives have been met (smoother traffic flow, journey time reliability) others have not (no increase in peak throughput, unable to suppress shock waves). As traffic conditions are difficult to replicate (e.g. day of the week, weather) it becomes challenging to perform like-for-like comparisons in the real world, and therefore hard to infer a direct benefit for a specific scenario. However, it seems generally accepted that such traffic control measures have benefits in smoothing traffic flows post accidents, and limit recurrence of congestion. This raises two points of interest specific to the simulation framework adopted in this paper. Firstly, that as the work is simulation based, like-for-like comparisons are feasible, as the same simulation conditions can be recreated many times. Secondly, that the unpredictability of human reaction and compliance is removed. Although institution obligations do not necessarily have to be obeyed, for variable speed limit compliance this could be more rigidly enforced, and thus identify what degree of compliance is required for the mechanism to achieve the intended effect, for example. In this case, the question of whether assessment of real world flow results is based on drivers complying with the speed limit is removed, and instead we see a more true (or arguably, idealized) view of what the impact would be.

Whilst the obligation received from the institution may not necessarily be obeyed, they can be considered as guidance for what to do in a given set of circumstances[2, 5, 22]. As such, this work also has a relationship with the field of collaborative behaviour between agents. Earlier scenarios focussed on convoy management using vehicle proximity data: we now considered this as an institutionally managed activity, in contrast to other coordination approaches (e.g. [27]). There is the aspiration that similar benefits can be demonstrated by self-organising vehicle collectives [15] for improved traffic flow and fuel savings via the institutional approach.

Having introduced the context and motivation for this work, the simulation framework which has been constructed to investigate the vehicle scenarios is now presented.

## 3. SIMULATION FRAMEWORK

The simulation framework has been designed with distribution

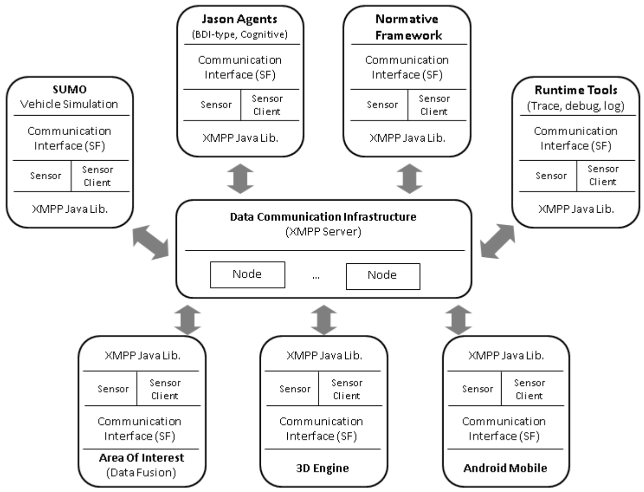



**Figure 1: Bath Sensor Framework (BSF) overview**

and a de-coupled approach to system component interoperability in mind [20]. The Extensible Messaging and Presence Protocol (XMPP) includes support for a publish-subscribe mechanism, allowing simulation members to publish without an overhead of managing consumers. Combined with the Resource Description Framework (RDF) specification, this data then includes a level of semantic annotation, providing subscribers with both the data and a definition along with it. Built around this, the Bath Sensor Framework (BSF) provides an 'out of the box' capability (opensource at http://code.google.com/p/bsf/) with various components based on the publish-subscribe approach. Some of these are fairly generic in nature (a database logger for RDFs, a replay tool to recreate events from database logs, performance testing tools), whereas others are more specific to the vehicle scenarios explored here (a 3D world view tool based on OpenStreetMap data, the Jason BDI engine).

The BSF aims to be a generic framework that has also been used to support intelligent agents controlling avatars in Second Life [21] and retro-fitted to the football scenario irst described in [26] (also in Second Life), as well as supporting the real-time collection and presentation of sensor data [11]. Earlier work on traffic simulation based on the BSF [3], presented a number of scenarios exploring communication between vehicles when acting as a convoy, investigating acceptable convoy performance while reducing the inter-vehicle communication, based on varying strategies. Discussion of new scenarios follow in the next section, but there have been substantial developments specific to the simulation framework. A schematic presentation of the BSF modules supporting the work described here is given in Figure 1.

Vehicle simulation is now performed by the 'Simulation of Urban MObility' (SUMO) [19] package, whereas previously the simulation was limited in terms of individual vehicle simulation, adherence to road networks and their rules, as well as general traffic representation. Through the use of a Java API, vehicle information is extracted and published to BSF subscribers, and a number of vehicle control commands have been implemented such that Jason agents are able to interact with and control SUMO vehicles. Furthermore, the richer simulation information provided by SUMO has allowed more investigations around the concepts of Situational Awareness discussed earlier.

One specific scenario based on this involves reasoning about traffic light data and how light states might impact the future state of the vehicle. Drawing from Endsley's 'projection' component, the consideration around how future events will effect an individual vehicle requires far greater computation. For this reason, a new simulation component referred to as the 'Area Of Interest (AOI)' module has been created. This can be considered as a data fusion engine, subscribing to data published by SUMO, calculating a vehicles AOI volume (based on current location and speed), and then publishing AOI RDF data back to the framework. Furthermore, as SUMO is handling vehicle routes, additional reasoning can be done based on what will be encountered in this AOI volume based on the current route, for example publishing upcoming traffic lights not just in the AOI in general, but that control lanes along the vehicle's route. This allows Jason agents to be able to react to both low level percepts (e.g. `+info(PosX, PosY, PosZ, Health, Heading)`) as well as much higher level (e.g. `+upcomingTrafficLight(Colour, Distance)`).

However, this improvement in simulation richness also introduces new challenges for the simulation framework itself. It was discussed in previous work [3] that there were performance differences dependent on the message volume, although scenarios at that point were quite lightweight in terms of data demand (four vehicles with one second update rate; four RDF messages per second). It was also found that when additional (agent mind state) data was broadcast from Jason, the resulting increase of up to approximately forty messages per second caused system instability. The introduction of SUMO leads to the possibility of simulating background traffic with potentially an increase in message volume by a factor of one hundred from the four vehicle scenarios used earlier. There is further complexity, as vehicles now need to exchange route information, and convey their light state (e.g. indicating, flashing lights, braking) and their performance metrics (e.g. fuel consumption, CO2 emission). There is also environmental information to be exchanged, such as traffic light states and flow detectors.

Consequently, there was a significant reworking of the simulation framework, to ensure that it is capable of meeting this requirement. An alternative XMPP message server (ejabberd) has been adopted, which yields significantly improved message throughput. Coupled with general code improvements, the system is comfortably handling 800 messages a second (over a wireless network). Due to the importance of message delivery in this framework, part of the build test now performs checks for message loss and message transfer rate in order to ensure the deployed hardware and network configuration performance is acceptable.

Ameliorating the bottleneck in message delivery now reveals costs in the data serialization task. As previously mentioned, clients transform data into RDF triples before publishing, but this is relatively expensive. Where semantic annotation is not needed JSON provides an efficient alternative format, that has been measured as significantly quicker than RDF due to the improved serialization performance (along with a wider range of benchmarks [28]). This bears relevance to the knowledge transfer theme, as it suggests the possibility to transfer at high volume rates but with little semantic description (i.e. JSON), or at low volume rates but with additional semantic definitions and analysis possible (i.e. RDF). In general we consider this a problem of impedance matching; that publishers and subscribers need to be matched not just in terms of data rates but also knowledge richness. For example, it has been found that there are issues if publishing at high rates to the Jason BDI engine, and that a lower rate of richer data is more suitable. Conversely, the 3D viewer is better suited to high rate, low level information (e.g. position updates) and not so well placed to display high level information (at least, in a raw format).

# 4. INSTITUTIONS

We are motivated to incorporate institutional reasoning into the simulation framework for two immediate reasons: (i) the breadth of possible situations requiring resolution between vehicles is too great to encode prior to runtime, and (ii) to be able to constrain a vehicle's sole pursuit of its own goals in order to consider the greater society of vehicles, through the enforcement of some form of global obligations.

In the first case, a scenario based on a somewhat ambiguous situation has been chosen: a vehicle becomes obstructed by a slower moving vehicle and wishes to get past. To indicate this, the vehicle flashes its front lights, and if this cue is interpreted correctly by the leading vehicle, it would change lanes in order to yield to the other vehicle's desire. With the simulation framework outlined in Section 3, the Jason agent is able to refer its requirement to the institution, updating the institution manager with the event `flashLights(Agent)` which in turn generates the institutional event `iniOblChangeLane(Agent)`. This in turn generates the obligation `obl(changeLane(Agent)` which the institution manager packages as an RDF triple and transmits to the BSF. As Jason agents are subscribed to the institution node, they receive this obligation, resulting in that agent's belief base being updated with the percept `+changeLane`, for which the agent can then decide to issue a command to its SUMO vehicle to move to different lane.

In the second case, an institution was defined to handle information regarding upcoming traffic lights, and issuing appropriate obligations to ensure the vehicle arrives at that light when it is green, rather than being held at a red light[1]. In this scenario, traffic light information is received via RDFs from the Area of Interest module, and where the distance to an upcoming light is between 100m to 300m and that light is red, the institution is updated with the event `upcomingRedLight(Agent)`. This then generates the institutional event `iniOblSlowDown(Agent)`, resulting in the obligation `obl(reduceSpeed(Agent)`, which the Jason agent receives and implements this by reducing its speed for a specified (35 second) period.

We demonstrate the impact of such institutional obligations in the experiments that follow.

As noted earlier, such behaviours could easily be encoded directly, if they were considered as part of the requirements, but this necessitates both fore-knowledge of the requirement and that it is fixed. We regard the mixed driving scenario as one example of the rich variety of socio-cognitive systems, populated by humans and software, mediated by technological artefacts, that are now emerging, where new requirements arise over time and old requirements change, rendering conventional software engineering approaches obsolete. Institutions are one way to provide a form of late binding of behaviour in order to address this issue. As also noted elsewhere, multiple institutions, while inevitably risking the creation of conflicting obligations [23], further enrich the environment, while keeping knowledge separated but linked [12].

# 5. EXPERIMENTAL SCENARIOS

Initial work using this framework focussed on information exchange in vehicle convoys, and explored the impact of various communication strategies on convoy performance. However, as discussed in the previous section, there were some limitations on what was simulated in those scenarios. Now with the improved vehicle and traffic simulation, coupled with improved message transfer capability, more advanced scenarios have been constructed.

Focus is on the use of a normative framework to control aspects

[1]Inspired by the earlier cited Audi news item [30]

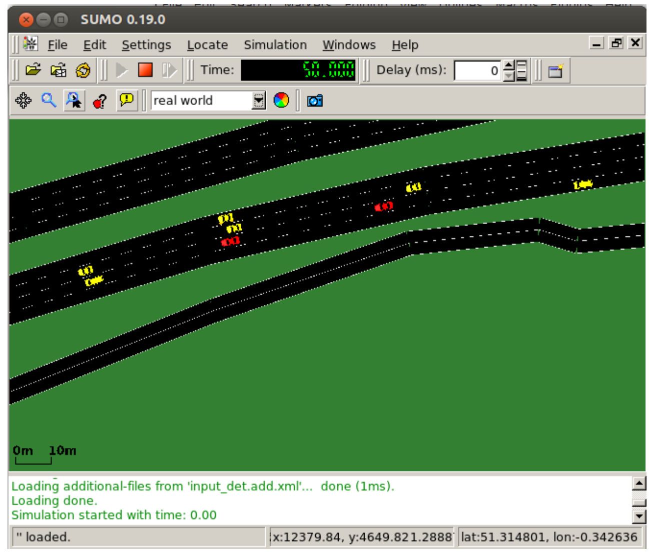



**Figure 2: Scenario 1 M25 Motorway**

of the vehicles, motivated by two factors. Firstly, while the BDI approach has been found to be robust in the vehicle domain, a combination with an institutional framework provides greater flexibility. The BDI agents are able to perform plan selection based on their belief state, with the added layer of the institution acting as a late binding mechanism, able to influence the agents ultimate approach. Secondly, the BDI agents are vehicle-centric in their view of goal achievement (i.e. pursuing their individual goals without concern regarding benefits for the society of vehicles), and the use of the institution model allows us to introduce a more society-centric consideration. For example, the institution can issue obligations to slow down to vehicles, in order to improve traffic flow for the greater population of vehicles, a method already in use via variable speed limits, as discussed earlier in Section 2.

Due to this shift in scenario focus, results no longer focus purely on underlying message metrics and convoy cohesion. Instead, there are now measurements of individual vehicle performance metrics (e.g. fuel consumption, emissions), as well as global metrics (e.g. flow volume, average speeds). Videos of scenario runs are also made available.

Two scenarios have been constructed to explore the impact of the introduction of the institution framework, which are now discussed in more depth.

## 5.1 Scenario 1: 'Move out of way' obligation

This scenario explores the ability of the institution to issue obligations based on the needs of other users in the road system, which may be contrary to the desire of individual vehicles. Currently, this is demonstrated through the use of two vehicles along a section of the M25 motorway in the UK. Flow traffic has been populated in SUMO, based on data from the UK Highways Agency Traffic Flow Database System (TRADS [1]), for this road section in order to provide a representation of background traffic flow. A snapshot of the scenario can be seen in Figure 2 where background traffic is yellow vehicles, and Jason controlled vehicles are red.

In the scenario context shown in Figure 2, a leading vehicle (V1) is travelling slightly slower than a trailing vehicle (V2), and as V2 wishes to maintain its speed without changing course (i.e. moving to another lane to overtake) it gains on V1. As the vehicles near each other, the Area Of Interest (AOI) module informs vehicles of other vehicle locations in their AOI volume. The Jason agents then perform a finer granularity check using their perceived collision volume space (directly in front), and determine whether another vehicle is in this space, along with distance to that vehicle. As V2 gains on V1, this behaviour is triggered, and if V1 is between 60m to 40m ahead, V2 will flash its lights at V1. With the institution running, V1 is issued an obligation to change lane, and perform this request. Without the institution, V2 will continue to gain on V1, and below 40m V2 will brake hard in order to avoid a collision. In this case, once V1 is detected as leaving the collision volume, V2 increases speed again, and a cyclical catch up – slow down behaviour is expected.

Such a scenario has applications elsewhere in the road domain, for example an emergency vehicle can create a similar requirement to move past.

## 5.2 Scenario 2: React to likely future events

This scenario explores the ability of agents to reason about future states of the environment in which they operate. Specifically, given a current route, what bearing the future state of traffic lights will have on that agent.

In this case, similarly to the previous scenario, the AOI module detects any traffic lights within the AOI volume. Upon detection, a route analysis determines whether any of these traffic lights control a lane on that route. If so, then the institution is informed about the traffic lights current colour state, and the distance to that light. Based on this, the institution is able to issue an obligation to reduce speed, so as to arrive at that light when it is green rather than red.

As discussed earlier, there are some similarities to the system produced by Audi [30]. However, in the scenario implemented here, the speed modification is enforced in order to assess the impact on the larger vehicle population, as well as the individual vehicle. Parallels can be seen with mechanisms such as variable speed limits on motorways discussed previously.

The route taken in this scenario is shown in Figure 3, with some annotation added to explain key areas. The 'START' and 'END' locations correspond to the area on the map where the vehicle is inserted, and location when the simulation is finished. The numerical labels refer to the three junctions controlled by traffic lights located along this route.

The results of these scenarios are now presented.

# 6. RESULTS

This section presents results for the two scenarios discussed in the previous section, comprising of a baseline without institution involvement, and with institution issued obligations.

## 6.1 Scenario 1

In this scenario, there were two configurations for the experiments. The first, with the institution inactive, involves vehicle 2 approaching vehicle 1 until a distance threshold triggers a hard brake (in order to avoid a collision). Once vehicle 1 has left the collision volume, vehicle 2 returns to the previous speed and so begins to gain on vehicle 1 again. The second, is with the institution active, which issues an obligation to vehicle 1 to change lane before the need for a sudden brake occurs.

In Figure 4 we can see the fuel consumption profiles of the two Jason controlled vehicles in this scenario, with the two variations of having the institution active and inactive. To clarify, V1 produces the same result with and without the institution, V2 only shows variation between 45 to 50 seconds. In both cases, Vehicle 2 has a slightly higher fuel consumption rate, as it is travelling at a higher

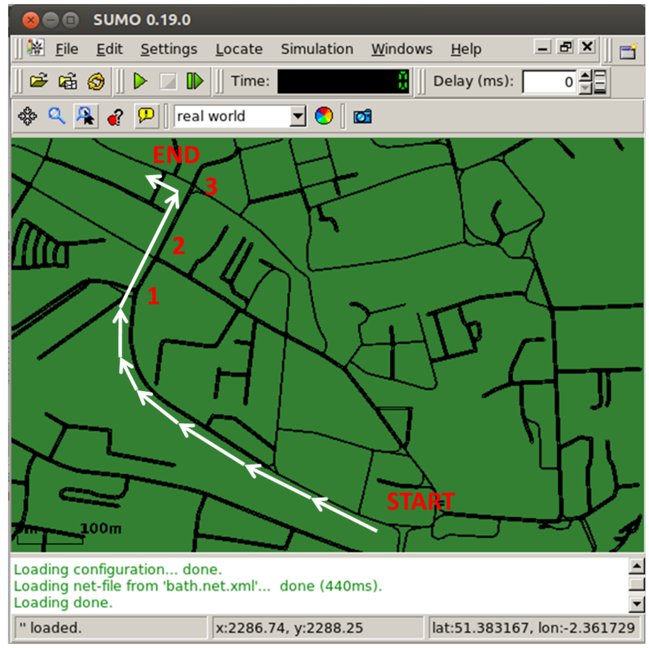



**Figure 3: Scenario 2 Bath City Centre**

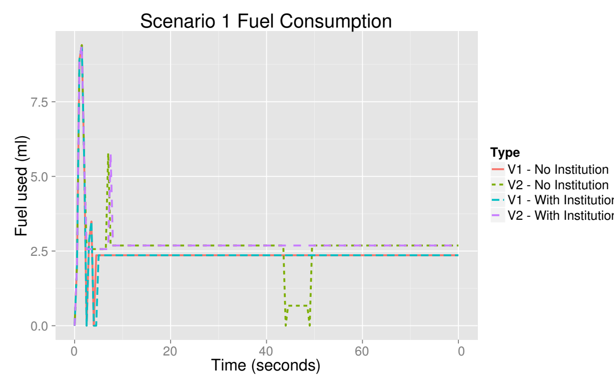



**Figure 4: Scenario 1 Fuel consumption comparison**

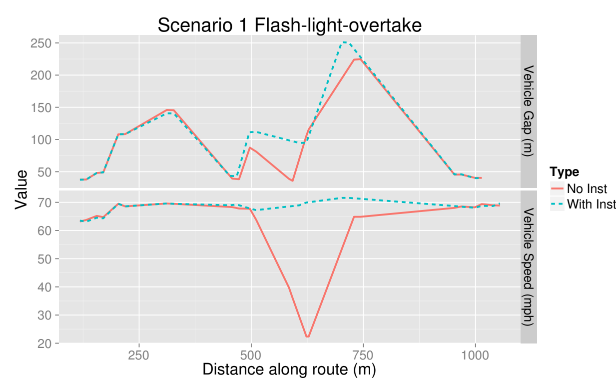



**Figure 5: Scenario 1 Vehicle speed and gaps along route**

speed than Vehicle 1. Ignoring initial fluctuations (as the simulation moves to steady state), we can see the main, and only, perturbation occurs in V2 at approximately 45 seconds. This is the point where it has got too close to the vehicle ahead (as there is no response to flashing its headlights) and has to brake. After about 5 seconds the vehicle ahead has moved out of its collision zone, and resumes its previous speed.

By comparison, with the institution issuing the obligation to change lane, the need to reduce speed is removed, and as such the fuel consumption profile remains constant. Fuel consumption is currently implemented in SUMO based on the Handbook Emission Factors for Road Transport (HBEFA) model, and with the motorway scenario it has been found that there is largely linear correlation between speed and fuel consumption. As such, the impact of excessive braking and acceleration is not captured in the fuel metric, however there is development effort under way to implement an alternative model in SUMO, which would result in more realistic – and, for the institutionally governed experiment, improved – figures for this scenario.

With the background traffic flow present, there is the desire to measure a more global metric rather than focussing on individual vehicles, in order to ascertain the impact of the behaviour of V1 and V2 on the general population. In order to achieve this, each vehicle reports its position along the scenario route along with its current speed and distance to the vehicle ahead. This provides an indication of disruption to the average speed (i.e. a vehicle having to slow down) and congestion (i.e. vehicles close together).

The results reported by this set of measurements can be seen in Figure 5, with results of the scenario with and without the institution involvement. The clearest result here is shown in the bottom graph, where with no institution running the vehicle speed at approx 600m along the route drops to nearly 20mph. This has occurred where V2 had to brake after getting too close to V1, and so expected vehicle speeds at this point in the route are affected. The upper graph of vehicle gaps is less conclusive in this particular scenario. There is clearly some difference when the institution is active, which could be as V1 changed lane, the gap ahead of V2 is now measured to the vehicle which was ahead of V1, and so we see a different profile here. Further work is planned to refine these measurements, and to incorporate other lanes (e.g. as V1 changes lane, what impact to we see in the lane it moves into).

## 6.2 Scenario 2

In this scenario, two experimental variations are reported. Firstly, a baseline where the vehicle is given its route, and SUMO handles speed control. In this case, the vehicle will obey the appropriate speed restriction for that road, and slow down, if required to, for events such as turns at junctions.

It can be seen in the 'no institution' results of Figure 6 that there is a significant variation in fuel consumption usage. The vehicle initially accelerates to the appropriate speed for that road, with its fuel use remaining constant until it arrives at junction 1, which is on a red light. The vehicle comes to a stop at this light and idles for five seconds, until the light turns green. The vehicle then accelerates and arrives at the second junction which is also on a red light. This light changes to green before the vehicle starts to idle, at which point the vehicle accelerates again and arrives at the third junction. The vehicle slows as this is a left hand turn, before reaching the 'END' location at approximately ninety seconds.

In comparison, the 'with institution' results in Figure 6 show a clearly different profile, due to the different chain of events caused by the institution involvement. In this case, at 15 seconds the institution issues the obligation `obl(reduceSpeed(Agent))`, resulting in the vehicles speed being reduced for 35 seconds. At 55 seconds this action is completed, and so the vehicle increases its speed back to the road limit. However, the vehicle arrives at junction 1 while the light is green, and so does not waste fuel idling or having to perform acceleration from stationary. The vehicle then passes through junction 2 as well, and at 85 seconds reduces speed for the turning at junction 3. This is followed by a spike in fuel consumption to increase speed, and approximately 95 seconds the vehicle arrives at the 'END' location.

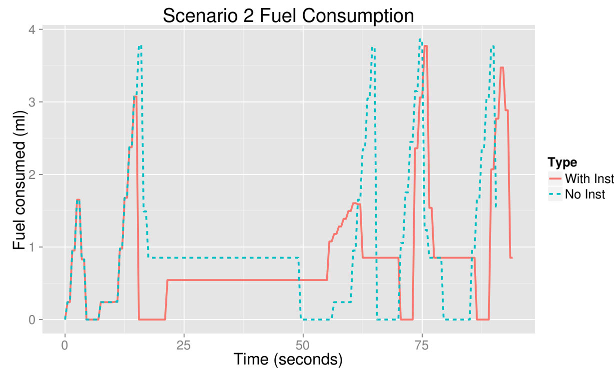



**Figure 6: Scenario 2 fuel consumption comparison**

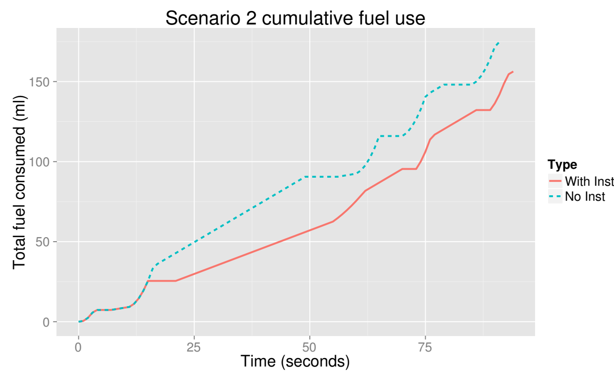



**Figure 7: Cumulative fuel consumption**

Whilst the individual fuel consumption profiles are useful to explore in relation to events in each scenario, a more substantial result can be found when taking the cumulative fuel consumptions for non institution and institution variants of this scenario. The results of this are shown in Figure 7.

Here a direct comparison of the results presented in Figure 6 can more easily be made, and there are some key findings to draw out. Firstly, despite the fact that the institution has enforced a slower speed on the vehicle for a significant duration of the scenario, the vehicle arrives at almost the same time (3 seconds difference) in both variations. However, in the institution variant of the scenario, there is almost 20ml less fuel used, approximately 10 percent less. As there is a correlation between emissions and fuel consumption, this also signifies that there is a significant reduction in $CO_2$.

# 7. DISCUSSION AND FUTURE WORK

In essence this work is an investigation into knowledge representation and transmission across a distributed platform, set in the context of intelligent vehicle systems. By performing elements of data fusion, and allowing components within the system to subscribe to their desired information source, we explore the question of whether it becomes possible to understand more, but communicate less. This ethos carries through the various system components, for example the ability of the 3D viewer to represent both environment spatial information through to agent mind state beliefs and plans.

Specific to the use of norms, the aspiration is to reduce the burden of coding for every eventuality, by aggregating data to suitable levels and triggering more powerful plans and actions based on this. Rather than Jason agents having to reason about their physical spatial state in relation to upcoming traffic lights, they receive more appropriate belief updates at a higher level. Similarly the institution does not have to micro-manage vehicle speed, instead it issues a higher level obligation to slow down, and leaves this to the vehicle to resolve appropriately.

To explore the benefits of such functionality, the two scenarios demonstrate areas where human drivers struggle with uncertainty in the selection of appropriate actions, both for their own benefit, and (with even more difficulty) what to do for the greater collective benefit. In these cases, a single institution has been shown in each scenario as being capable of resolving, and improving, the situation.

Results from the first scenario of a vehicle moving out of the way of another vehicle show a clear variation in fuel consumption, though a less clear overall impact of this to the wider vehicle population. The results generated so far highlight a localised decrease in speed, and some impact on gaps between vehicles. However, further refinement is needed in order to identify factors such as the number of vehicles affected and duration of the disruption. Furthermore, the new fuel consumption model planned for SUMO will be used to reassess the fuel consumption expectations of the excessive brake-accelerate behaviour in this scenario.

Results from the second scenario show a benefit to the individual vehicle adopting the institutions obligations. By reducing that vehicles speed (to the detriment of the apparent benefit of arriving at its destination faster) both fuel consumption and emissions are reduced. This scenario will also be expanded to include background traffic, as well as the complexity of how to manage multiple traffic lights, which may then become a minimisation problem (also suitable for resolution by some software component).

Having produced results which indicate there is a useful role as well as quantifiable benefit for institutions in governing a future of autonomous vehicles, further experimentation is planned. A scenario of global vs local Variable Speed Limit implementation (e.g. in managing congestion following the excessive braking of scenario 1) is currently being implemented, as well as post accident management (e.g. lane one vehicles required to merge with lane two which will provide richer scenarios from which to assess the benefit of multiple institution interactions.

## 8. REFERENCES


[1] UK Highways Agency. Traffic flow database system. Accessible via https:/trads.hatris.co.uk. accessed 26th Jan 2014.

[2] N. Alechina, M. Dastani, and B. Logan. Programming norm-aware agents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '12, pages 1057–1064, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[3] V. Baines and J. Padget. Communication and metrics in agent convoy organization. In *7th International Workshop on Agents in Traffic and Transportation (ATT 2012 at AAMAS 2012)*, pages 69–77, June 2012.

[4] T. Balke. *Towards the Governance of Open Distributed Systems: A Case Study in Wireless Mobile Grids*. PhD thesis, University of Bayreuth, September 2011.

[5] T. Balke, M. De Vos, J. Padget, and D. Traskas. On-line reasoning for institutionally-situated BDI agents. In *The 10th International Conference on Autonomous Agents and Multiagent Systems*, volume III, pages 1109–1110, May 2011.

[6] C. Bergenhem, Q. Huang, A. Benmimoun, and T. Robinson. Challenges of platooning on public motorways. In *17th World Congress on Intelligent Transport Systems*, 2010.

[7] Carl Bergenhem, Erik Hedin, and Daniel Skarin. Vehicle-to-vehicle communication for a platooning system. *Procedia - Social and Behavioral Sciences*, 48(0):1222 – 1233, 2012.

[8] Katrin Bilstrup, Annette Böhm, Kristoffer Lidström, Magnus Jonsson, Tony Larsson, Lars Strandén, and Hossein Zakizadeh. Vehicle alert system. In *Proceedings of 14th world congress on intelligent transport system (ITS)*, pages 2–9, 2007.

[9] R. H. Bordini, J. F. Hübner, and M Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*. Wiley, 2007.

[10] M. E. Bratman, D. J. Israel, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988.

[11] Teresa Chiang, Gokhan Mevlevioglu, Sukumar Natarajan, Julian Padget, and Ian Walker. Inducing [sub]conscious energy behaviour through visually displayed energy information: A case study in university accommodation. *Energy and Buildings*, 70(0):507 – 515, 2014.

[12] O. Cliffe, M. De Vos, and J. A. Padget. Specifying and reasoning about multiple institutions. In *Coordination, Organizations, Institutions, and Norms in Agent Systems Lecture Notes in Artificial Intelligence*. Springer, 2007.

[13] K. Dresner and P. Stone. Sharing the road: Autonomous vehicles meet human drivers. In Manuela M. Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, January 2007.

[14] Mica R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.

[15] Sándor P. Fekete, Björn Hendriks, Christopher Tessars, Axel Wegener, Horst Hellbrück, Stefan Fischer, and Sebastian Ebers. *Methods for Improving the Flow of Traffic*, chapter 5. Autonomic Systems. Birkhäuser Verlag, 2011.

[16] Freie Universitat Berlin. Autonomous car navigates the streets of Berlin. http://autonomos.inf.fu-berlin.de/news/press-release-92011, September 2011. accessed October 8th, 2011.

[17] Nikki Gordon-Bloomfield. Nissan takes Japanese PM on autonomous LEAF test drive. http://transportevolved.com/2013/11/11/nissan-takes-japanese-pm-on-autonomous-leaf-test-dr November 2013. accessed January 19th, 2014.

[18] Kyeong Tae Kim. STVC: Secure Traffic-light to Vehicle Communication. In *ICUMT*, pages 96–104. IEEE, 2012.

[19] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[20] JeeHang Lee, Vincent Baines, and Julian Padget. Decoupling cognitive agents and virtual environments. In Frank Dignum, Cyril Brom, Koen V. Hindriks, Martin D. Beer, and Deborah Richards, editors, *CAVE*, volume 7764 of *Lecture Notes in Computer Science*, pages 17–36. Springer, 2012.

[21] JeeHang Lee, Tingting Li, and Julian Padget. Towards polite virtual agents using social reasoning techniques. *Computer Animation and Virtual Worlds*, 24(3-4):335–343, 2013.

[22] JeeHang Lee, Julian Padget, Brian Logan, Natasha Alechina, and Daniela Dybalova. Run-time norm compliance in BDI agents. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 2014*. IFAAMAS, 2014. To appear.

[23] Tingting Li, Tina Balke, Marina De Vos, Julian Padget, and Ken Satoh. Legal conflict detection in interacting legal systems (JURIX 2013). In Kevin D. Ashley, editor, *Legal Knowledge and Information Systems*, volume 259 of *Frontiers in Artificial Intelligence and Applications*, pages 107–116. IOS Press, 2013.

[24] J. Markoff. Google cars drive themselves, in traffic. http://www.nytimes.com/2010/10/10/science/10google.html, October 2010. accessed October 8th, 2011.

[25] M. Papageorgiou, E. Kosmatopoulos, and I. Papamichail. Effects of variable speed limits on motorway traffic flow. *Transportation Research Record*, 2047(-1):37–48, 2008.

[26] S. Ranathunga, S. Cranefield, and M. Purvis. Interfacing a cognitive agent platform with a virtual world: a case study using Second Life (extended abstract),. In Sonenberg Tumer, Yolum and Stone, editors, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011),*, pages 1181–1182, 2011.

[27] Brahim Chaib-draa and Simon Hallé. A collaborative driving system based on multiagent modelling and simulations. *Journal of Transportation Research Part C (TRC-C): Emergent Technologies*, 13(4):320–345, 2005.

[28] Eishay Smith. JVM-Serializers. https://github.com/eishay/jvm-serializers/wiki, 2013. accessed February 2nd, 2014.

[29] Mehdi Fallah Tafti. An investigation on the approaches and methods used for Variable Speed Limit control. In *15th World Congress on Intelligent Transport Systems and ITS America's 2008 Annual Meeting*, 2008.

[30] Volkswagen. Audi in the simTD large-scale test study: the "traffic light info online" project. http://www.volkswagenag.com/content/vwcorp/info_center/en/news/2013/06/audi_simTD.html, June 2013. accessed January 19th, 2014.

[31] XMPP Standards Foundation. The XMPP standards foundation homepage. Retrieved from http://www.xmpp.org, 20130129, no date.

# Online Cost-Sharing Mechanism Design for Demand-Responsive Transport Systems*


Masabumi Furuhata
USC
Los Angeles, USA
k439bk439b@gmail.com

Kenny Daniel
Algorithmia Inc.
San Francisco, USA
kenny@algorithmia.io

Sven Koenig
USC
Los Angeles, USA
skoenig@usc.edu

Fernando Ordonez
Universidad de Chile
Santiago, Chile
fordon@dii.uchile.cl

Maged Dessouky
USC
Los Angeles, USA
maged@usc.edu

Marc-Etienne Brunet
McGill University
Montreal, Canada
marcetiennebrunet@gmail.com

Liron Cohen
USC
Los Angeles, USA
lironcoh@usc.edu

Xiaoqing Wang
USC
Los Angeles, USA
xiaoqinw@usc.edu



## ABSTRACT

Demand-responsive transport (DRT) systems provide flexible transport services for passengers that request door-to-door rides in shared-ride mode without fixed routes and schedules. One has to design cost-sharing mechanisms for offering fare quotes to potential passengers so that all passengers are treated fairly. The main issue is how the operating costs of the DRT system should be shared among the passengers (given that different passengers cause different amounts of inconvenience to the other passengers), taking into account that DRT systems should provide fare quotes instantaneously without knowing future ride requests. We propose a novel cost-sharing mechanism, called Proportional Online Cost Sharing (POCS), that provides passengers with upper bounds on their fares immediately after their arrivals, allowing them to accept their fare quotes or drop out. We then demonstrate that POCS has attractive properties for both shuttle providers and passengers.


## 1. INTRODUCTION

Demand-responsive transport (DRT) systems provide flexible transport services where individual passengers request door-to-door rides by specifying their desired start and end locations. Multiple shuttles service these requests in shared-ride mode without fixed routes and schedules. DRT services are more flexible and convenient for passengers than buses since they do not operate on fixed routes and schedules, yet are cheaper than taxis due to the higher utilization of transport capacity. In the United States, DRT services are commonly used to service the transport needs of disabled and elderly citizens and have experienced rapid growth, for example, in the form of dial-a-ride paratransit services mandated under the Americans with Disabilities Act, while the National Transit Summaries and Trends report that typical DRT systems are highly subsidized.

In this paper, we propose a novel cost-sharing mechanism, called Proportional Online Cost Sharing (POCS), that provides passengers with upper bounds on their fares immediately after their arrivals, allowing them to accept their fare quotes or drop out. We then demonstrate that POCS has attractive properties for both shuttle providers and passengers. How passengers should share the operating cost in an online setting, where knowledge of future ride requests is missing, is a non-trivial problem for the following reasons: First, passengers do not submit their ride requests at the same time but should be given incentives to submit their ride requests as early as possible to allow the DRT systems more time to find routing solutions that can offer subsequent passengers lower fares due to synergies with the early ride requests, which might allow them to service more passengers. Second, passengers have different start and end locations and thus cause different amounts of inconvenience to the other passengers, which should be reflected in the fares. Finally, passengers should be quoted fares immediately after submitting their ride requests. This gives passengers certainty about the cost of service and allows the DRT system to plan routes better knowing which passengers have committed to participate. This requires DRT systems to make instantaneous and irreversible decisions despite having no knowledge of future ride requests [2].

## 2. ONLINE COST SHARING

In this section, we define the online cost-sharing problem for demand responsive transport (DRT) systems, provide an example, discuss existing cost-sharing mechanisms and some of their shortcomings, and finally derive a list of desirable properties for online cost-sharing mechanisms for DRT systems.

### 2.1 Problem Definition

DRT systems provide flexible transport services where individual passengers request door-to-door rides. Multiple shuttles service these requests without fixed routes and schedules. Passengers share shuttles. For example, after a passenger has been picked up and before it is dropped off,

*A longer version of this paper is under submission to the IEEE Transactions on Intelligent Transportation Systems. This paper was therefore submitted to this non-archival workshop purely for the exchange of ideas. Our research was supported by METRANS grant 09-19 via the California Department of Transportation and follow-up contract DTFH61-10-C-00030 from the Federal Highway Administration under the Broad Agency Announcement of Exploratory Advanced Research (EAR).

other passengers can be picked up and dropped off, resulting in a longer ride for the passenger. Passengers need to pay a share of the operating cost. Passengers arrive (that is, submit their ride requests) one after the other by specifying their desired start and end locations. The arrival time of a passenger is the time when it submits its ride request. In case the passenger decides to delay its arrival, we distinguish its truthful arrival time, which is its earliest possible arrival time, from its actual, perhaps delayed, arrival time. We assume, for simplicity, that all passengers arrive before the shuttles start to service the passengers. We also assume, without loss of generality, that exactly one passenger arrives at each time $k = 1, \ldots, t$, namely that passenger $\pi(k)$ arrives at time $k$ under arrival order $\pi$, where an arrival order is a function that maps arrival times to passengers.

Definition 1. *For all times $k$ and all arrival orders $\pi$ with $1 \le k$, the alpha value $\alpha_{\pi(k)}$ of passenger $\pi(k)$ quantifies the demand of its request, that is, how much of the transport resources it requests. We assume that it is positive and independent of the arrival time of the passenger.*

These assumptions are, for example, satisfied for the shortest point-to-point travel distance from the start location to the end location of a passenger, which is the quantity that we use in this paper as its alpha value.

Definition 2. *For all times $t$ and all arrival orders $\pi$ with $1 \le t$, the total cost $totalcost^t_\pi$ at time $t$ under arrival order $\pi$ is the operating cost required to service passengers $\pi(1), \ldots, \pi(t)$. We define $totalcost^0_\pi := 0$ and assume that 1) the total cost is non-decreasing over time, that is, for all times $t$ and $t'$ and all arrival orders $\pi$ with $t \le t'$, $totalcost^t_\pi \le totalcost^{t'}_\pi$; and 2) the total cost at time $t$ is independent of the arrival order of passengers $\pi(1), \ldots, \pi(t)$, that is, for all times $t$ and all arrival orders $\pi$ and $\pi'$ with $1 \le t$ and $\{\pi(1), \ldots, \pi(t)\} = \{\pi'(1), \ldots, \pi'(t)\}$, $totalcost^t_\pi = totalcost^t_{\pi'}$.*

These assumptions are, for example, satisfied for the minimal operating cost, which is the quantity that we use in this paper for the total cost. The DRT system can accommodate advanced features, such as operating times and capacities of shuttles and time constraints of passengers, as long as it can determine total costs that satisfy the assumptions. The assumptions are typically not satisfied if passengers can arrive after the shuttles have started to service passengers since the shuttle locations influence the total cost. We initially assume for simplicity in the theoretical part of this paper that the DRT system can easily calculate the total cost at any given time.

Definition 3. *For all times $k$ and all arrival orders $\pi$ with $1 \le k$, the marginal cost $mc_{\pi(k)}$ of passenger $\pi(k)$ under arrival order $\pi$ is the increase in total cost due to its arrival, that is, $mc_{\pi(k)} := totalcost^k_\pi - totalcost^{k-1}_\pi$.*

Definition 4. *For all times $k$ and $t$ and all arrival orders $\pi$ with $1 \le k \le t$, the shared cost $cost^t_{\pi(k)}$ of passenger $\pi(k)$ at time $t$ under arrival order $\pi$ is its share of the total cost at time $t$.*

The DRT system provides a (myopic) fare quote to a passenger immediately after its arrival. The fare quoted to passenger $\pi(k)$ immediately after its arrival at time $k$ is $cost^k_{\pi(k)}$.

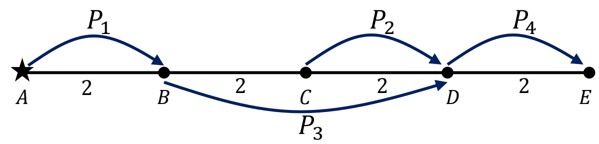



**Figure 1: DRT Example 1**

**Table 1: DRT Values**

| | | $k=1$ $\pi(k)=P_1$ | $k=2$ $\pi(k)=P_2$ | $k=3$ $\pi(k)=P_3$ | $k=4$ $\pi(k)=P_4$ |
|---|---|---|---|---|---|
| Alpha Value: | $\alpha_{\pi(k)}$ | 2 | 2 | 4 | 2 |
| Total Cost: | $totalcost^k_\pi$ | 40 | 120 | 120 | 160 |
| Marginal Cost: | $mc_{\pi(k)}$ | 40 | 80 | 0 | 40 |

(A fare quote of infinity means that the passenger cannot be serviced.)

Definition 5. *For all times $k$ and all arrival orders $\pi$ with $1 \le k$, the fare limit $w_{\pi(k)}$ of passenger $\pi(k)$ is the maximum amount that it is willing to pay for its requested ride.*

Passenger $\pi(k)$ drops out and is not serviced if its fare limit $w_{\pi(k)}$ is lower than its fare quote, that is, $w_{\pi(k)} < cost^k_{\pi(k)}$. In this case, the DRT system simply pretends that the passenger never arrived, which explains why we assume, without loss of generality, that all passengers accept their fare quotes. When the passenger accepts its fare quote and is serviced, its fare is $cost^t_{\pi(k)}$ (which is not guaranteed to equal its fare quote).

## 2.2 Demand-Responsive Transport Example

We use the DRT example in Figure 1 to illustrate typical cost-sharing mechanisms. There is one shuttle that can transport up to four passengers and starts at the star. The shuttle incurs an operating cost of 10 for each unit of distance traveled and needs to return to its initial location. There are four passengers with arrival order $\pi(1) = P_1$, $\pi(2) = P_2$, $\pi(3) = P_3$ and $\pi(4) = P_4$. For example, Passenger $P_3$ requests a ride from location B to location D, as shown in Figure 1. All passengers accept all fare quotes. Table 1 shows the alpha value of each passenger, the total cost after the arrival of each passenger and the marginal cost of each passenger. For example, the alpha value of Passenger $P_3$ is the shortest point-to-point travel distance from its start location B to its end location D. Thus, $\alpha_{\pi(3)} = 4$. The total cost at time 3, after the arrival of Passenger $P_3$, is 10 times the minimal travel distance of the shuttle required to service Passengers $P_1$, $P_2$ and $P_3$ and return to its initial locations. Thus, $totalcost^3_\pi = 120$ since the shuttle has to drive from location A (to pick up Passenger $P_1$) via location B (to drop off Passenger $P_1$ and pick up Passenger $P_3$) and location C (to pick up Passenger $P_2$) to location D (to drop off Passengers $P_2$ and $P_3$) and to return to its initial location A. The marginal cost of Passenger $P_3$ is the increase in total cost due to its arrival. Thus, $mc_{\pi(3)} = totalcost^3_\pi - totalcost^2_\pi = 120 - 120 = 0$ since the total cost remains 120.

## 2.3 Typical Cost-Sharing Mechanisms

Online cost-sharing mechanisms determine the shared costs in an online setting, where knowledge of future arrivals

**Table 2: Proportional Cost Sharing:** $cost^t_{\pi(k)}$

| | $k=1$ $\pi(k)=P_1$ | $k=2$ $\pi(k)=P_2$ | $k=3$ $\pi(k)=P_3$ | $k=4$ $\pi(k)=P_4$ |
|---|---|---|---|---|
| $t=1$ | 40 | | | |
| $t=2$ | 60 | 60 | | |
| $t=3$ | 30 | 30 | 60 | |
| $t=4$ | 32 | 32 | 64 | 32 |

of passengers is missing. We present typical cost-sharing mechanisms and some of their shortcomings in an online setting using the DRT example in Section 2.2.

### 2.3.1 Proportional Cost Sharing

One commonly used cost-sharing mechanism is *proportional cost sharing* [16, 14], where the total cost is distributed among all passengers proportionally to their alpha values, which reflects that passengers with higher demands should contribute more toward the total cost. Consequently, for all times $k$ and $t$ and all arrival orders $\pi$ with $1 \leq k \leq t$, the shared cost of passenger $\pi(k)$ at time $t$ under arrival order $\pi$ is

$$cost^t_{\pi(k)} := totalcost^t_\pi \frac{\alpha_{\pi(k)}}{\sum_{j=1}^t \alpha_{\pi(j)}}.$$

Instead of distributing the total (operating) cost among all passengers, one could also distribute the operating cost of each shuttle among all passengers serviced by that shuttle, which results in identical properties for the DRT example in Section 2.2 since there is only one shuttle in the DRT example.

Table 2 shows the shared costs for the DRT example. For example, the total cost at time 3 is 120. It is distributed among all passengers that have arrived by time 3, namely Passengers $P_1$, $P_2$ and $P_3$, proportionally to their alpha values, namely 2, 2 and 4, respectively. Consequently, the shared cost of Passenger $P_3$ at time 3 and thus the fare quoted to Passenger $P_3$ after its arrival is $cost^3_{\pi(3)} = 60$. Similarly, the total cost at time 4 is 160. It is distributed among all passengers that have arrived at time 4, namely Passengers $P_1$, $P_2$, $P_3$ and $P_4$, proportionally to their alpha values, namely 2, 2, 4 and 2, respectively. Consequently, the shared cost of Passenger $P_3$ at time 4 and thus its fare is $cost^4_{\pi(3)} = 64$, implying that its fare is higher than its fare quote at time 3. This is undesirable because Passenger $P_3$ might accept the fare quote but not the higher fare, meaning that it will have to drop out shortly before receiving its ride and then needs to search for a last-minute alternative to using the DRT system, which might be pricy and is not guaranteed to exist. Thus, we suggest that a fare quote should be an upper bound on the fare (*immediate-response property*). We also suggest that the upper bound should be reasonably low since passengers might otherwise look for alternatives to using the DRT system, commit to one and then drop out unnecessarily. Obtaining reasonably low upper bounds can be difficult since the DRT system has no knowledge of future arrivals of passengers.

### 2.3.2 Incremental Cost Sharing

Another commonly used cost-sharing mechanism is *incremental cost sharing* [9], where the shared cost of each passenger is its marginal cost, which is the increase in total cost due to its arrival. Consequently, for all times $k$ and $t$ and all arrival orders $\pi$ with $1 \leq k \leq t$, the shared cost of passenger $\pi(k)$ at time $t$ under arrival order $\pi$ is

$$cost^t_{\pi(k)} := mc_{\pi(k)}.$$

Table 3 (left) shows the shared costs for the DRT example in Section 2.2. For example, the marginal cost of Passenger $P_3$ is 0. Consequently, the shared cost of Passenger $P_3$ from its arrival at time 3 on is 0, and thus both its fare quote and fare are 0 as well. In general, incremental cost sharing satisfies the immediate-response property since the marginal costs are independent of time. The fares of Passengers $P_1$, $P_2$, $P_3$ and $P_4$ are 40, 80, 0 and 40, respectively. Thus, Passenger $P_3$ is a free rider, which is undesirable in general and especially in the context of the DRT example since Passenger $P_3$ has the highest demand, which should be reflected in the fares. Proportional cost sharing does not suffer from this problem. For the discussion below, notice that the fare per alpha value of Passenger $P_1$ is 20 and the one of Passenger $P_3$ is 0 even though Passenger $P_1$ arrives before Passenger $P_3$.

Table 3 (right) shows the shared costs for the DRT example in Section 2.2 if Passenger $P_1$ delays its arrival and the passengers arrive in order $P_2$, $P_1$, $P_3$ and $P_4$. Now, the shared cost of Passenger $P_1$ from its arrival at time 2 on is 0, and thus both its fare quote and fare are 0 as well. Thus, Passenger $P_1$ can reduce its fare from 40 to 0 by strategically delaying its arrival. This delay is undesirable because synergies with the early ride requests allow the DRT system to offer low fare quotes to new passengers. We therefore suggest to ensure that the best strategy of every passenger is to arrive truthfully (that is, as early as possible) because it cannot decrease its fare by delaying its arrival (*incentive-compatibility property*). Incremental cost sharing does not satisfy this property as shown above. Similarly, under incremental cost sharing, Passenger $P_1$ and $P_2$ prefers to pay the fare of Passenger $P_3$ rather than their own fare because Passenger $P_3$ enjoys a free ride due to payments of these two passengers. We therefore suggest that the fares per alpha value of passengers are never higher than those of passengers that arrive after them (*online-fairness property*).

## 2.4 Desirable Properties

None of the cost-sharing mechanisms discussed so far are well-suited for the DRT problem. Based on their shortcomings, we derive a list of desirable properties for online cost-sharing mechanism. Our primary objective is to design an online cost-sharing mechanism that provides incentives for passengers to arrive truthfully while satisfying basic properties of cost-sharing mechanism in general, such as fairness and budget balance.

**Online Fairness:** The shared costs per alpha value of passengers are never higher than those of passengers who arrive after them, that is, for all times $k_1$, $k_2$ and $t$ and all arrival orders $\pi$ with $1 \leq k_1 \leq k_2 \leq t$,

$$\frac{cost^t_{\pi(k_1)}}{\alpha_{\pi(k_1)}} \leq \frac{cost^t_{\pi(k_2)}}{\alpha_{\pi(k_2)}}.$$

**Immediate Response:** Passengers are provided immediately after their arrivals with (ideally low) upper bounds on their shared costs at any future time, that is, for all times $k$, $t_1$ and $t_2$ and all arrival orders $\pi$ with $1 \leq k \leq t_1 \leq t_2$,

$$cost^{t_1}_{\pi(k)} \geq cost^{t_2}_{\pi(k)}.$$

**Individual Rationality:** The shared costs of passengers who accepted their fare quotes never exceed their fare limits

**Table 3: Incremental Cost Sharing: $cost^t_{\pi(k)}$**

| | Truthful Arrival | | | | Delayed Arrival | | | |
|---|---|---|---|---|---|---|---|---|
| | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ |
| | $\pi(k)=P_1$ | $\pi(k)=P_2$ | $\pi(k)=P_3$ | $\pi(k)=P_4$ | $\pi(k)=P_2$ | $\pi(k)=P_1$ | $\pi(k)=P_3$ | $\pi(k)=P_4$ |
| $t=1$ | 40 | | | | 120 | | | |
| $t=2$ | 40 | 80 | | | 120 | 0 | | |
| $t=3$ | 40 | 80 | 0 | | 120 | 0 | 0 | |
| $t=4$ | 40 | 80 | 0 | 40 | 120 | 0 | 0 | 40 |

at any future time, that is, for all times $k$ and $t$ and all arrival orders $\pi$ with $1 \leq k \leq t$,

$$cost^t_{\pi(k)} \leq w_{\pi(k)}.$$

**Budget Balance:** The total cost equals the sum of the shared costs of all passengers, that is, for all times $t$ and all arrival orders $\pi$ with $1 \leq t$,

$$\sum_{j=1}^{t} cost^t_{\pi(j)} = totalcost^t_{\pi}.$$

**Ex-Post Incentive Compatibility:**[1] The best strategy of every passenger is to arrive truthfully, provided that all other passengers arrive truthfully as well and do not change whether they accept their fare quotes or drop out, because it then cannot decrease its shared cost by delaying its arrival, that is, for all times $k_1$, $k_2$ and $t$ and all arrival orders $\pi$ and $\pi'$ with $1 \leq k_1 < k_2 \leq t$ and

$$\pi'(k) = \begin{cases} \pi(k+1) & \text{if } k_1 \leq k < k_2 \\ \pi(k_1) & \text{if } k = k_2 \\ \pi(k) & \text{otherwise,} \end{cases}$$
$$cost^t_{\pi(k_1)} \leq cost^t_{\pi'(k_2)}.$$

The online fairness and ex-post incentive-compatibility properties are similar but one does not imply the other. Basically, they provide incentives for passengers to arrive truthfully. Thus, the DRT systems have more time to prepare and might also be able to offer subsequent passengers lower fares due to synergies with the early ride requests, which might allow them to service more passengers. The online-fairness property is also meant to ensure that passengers consider the fares to be fair. The immediate-response property enables DRT systems to provide fare quotes, in form of upper bounds on the fares, to passengers immediately after their arrivals despite missing knowledge of future arrivals of passengers. Thus, passengers have no uncertainty about whether they can be serviced or how high their fares will be, while the DRT systems reduce their uncertainty about passengers dropping out and can thus prepare better. Yet, the DRT system still retains some flexibility to optimize the routes and schedules after future arrivals of passengers. The budget-balance property guarantees that the sum of the fares of all passengers always equals the total cost. Thus, no profit is made and no subsidies are required.

We stated sufficient rather than necessary conditions for the properties. For example, the budget-balance property could be weakened to state that the total cost equals the sum of the shared costs of all passengers after the arrival of the last passenger. Requiring the properties to be satisfied at any time rather than only after the arrival of the last passenger simplifies the development of the online cost-sharing mechanism since they do not know in advance when the last passenger arrives.

[1]We would like the ex-post incentive-compatibility property ideally to state that the best strategy of every passenger is to arrive truthfully because it cannot decrease its shared cost by delaying its arrival. However, we impose two conditions in this paper that we hope to be able to relax in the future. The first condition is that all other passengers arrive truthfully, which, for example, rules out collusion of several passengers. In general, the literature on online-mechanism design [12] distinguishes two types of incentive compatibility, namely *dominant-strategy incentive compatibility* and *ex-post incentive compatibility.* Dominant-strategy incentive compatibility does not require the first condition, while ex-post incentive compatibility does. Dominant-strategy incentive compatibility is difficult to achieve in an online setting [12], which is why we impose the first condition in this paper. The second condition is that the other passengers do not change whether they accept their fare quotes or drop out, even though, for example, the delayed arrival of a passenger could cause the fare quotes of subsequent passengers to increase, which might make them drop out. The arrival orders with and without the delayed arrival of the passenger are then difficult to relate, which is why we impose the second condition in this paper.

# 3. POCS

In this section, we describe a novel online cost-sharing mechanism, called Proportional Online Cost Sharing (POCS), which satisfies the properties listed in Section 2.4, as proved in the technical report [4]. The idea behind POCS is the following: POCS partitions passengers into coalitions, where coalitions contain all passengers that arrive within given time intervals (rather than, for example, all passengers served by the same shuttle). Initially, each newly arriving passenger forms its own coalition. However, passengers can choose to form coalitions with passengers that arrive directly after them to decrease their shared costs per alpha value, which implies the online fairness, immediate response, and ex-post incentive-compatibility properties. For example, the immediate-response property is satisfied because passengers add other passengers to their coalitions only when this decreases their shared costs per alpha value and thus also their shared costs (since the alpha values are positive).

## 3.1 Calculation of Shared Costs

We now describe how POCS calculates the shared costs.

Definition 6. *For all times $k_1$, $k_2$ and $t$ and all arrival orders $\pi$ with $k_1 \leq k_2 \leq t$, the coalition cost per alpha value of passengers $\pi(k_1), \ldots, \pi(k_2)$ at time $t$ under arrival order $\pi$ is*

$$ccpa_{\pi(k_1,k_2)} := \frac{\sum_{j=k_1}^{k_2} mc_{\pi(j)}}{\sum_{j=k_1}^{k_2} \alpha_{\pi(j)}}.$$

Definition 7. *For all times $k$ and $t$ and all arrival orders $\pi$ with $k \leq t$, the shared cost of passenger $\pi(k)$ at time $t$*

*under arrival order $\pi$ is*

$$cost^t_{\pi(k)} := \alpha_{\pi(k)} \min_{k \le j \le t} \max_{1 \le i \le j} ccpa_{\pi(i,j)}.$$

## 3.2 Other Cost-Sharing Mechanisms

The following definition and lemma, whose proof is provided in the technical report [4], helps to understand the similarities between POCS and other cost-sharing mechanisms. It states that the shared costs per alpha value of all passengers in any coalition are always identical and equal to the coalition cost per alpha value of the coalition.

DEFINITION 8. *For all times $k_1$, $k_2$ and $t$ and all arrival orders $\pi$ with $k_1 \le k_2 \le t$, a coalition $(k_1, k_2)$ at time $t$ is a group of passengers $\pi(k_1), \ldots, \pi(k_2)$ with*

$$\frac{cost^t_{\pi(k)}}{\alpha_{\pi(k)}} = \frac{cost^t_{\pi(k_1)}}{\alpha_{\pi(k_1)}}$$

*for all times $k$ with $k_1 \le k \le k_2$ and the preceeding equality not holding for all times $k$ with ($k = k_1 - 1$ or $k = k_2 + 1$) and $1 \le k \le t$.*

LEMMA 1. *The shared cost per alpha value of any passenger in any coalition at any time equals the coalition cost per alpha value of the coalition, that is, for all times $k_1$, $k$, $k_2$ and $t$ and all arrival orders $\pi$ with $1 \le k_1 \le k \le k_2 \le t$ such that $(k_1, k_2)$ is a coalition at time $t$,*

$$\frac{cost^t_{\pi(k)}}{\alpha_{\pi(k)}} = ccpa_{\pi(k_1,k_2)}.$$

Lemma 1 implies that POCS is a combination of proportional and incremental cost sharing. The sum of the marginal costs of all passengers in any coalition ("the total cost of all passengers in the coalition") at time $t$ is distributed among all passengers in the coalition proportionally to their alpha values since, for all times $k_1$, $k$, $k_2$ and $t$ and all arrival orders $\pi$ with $k_1 \le k \le k_2 \le t$ such that $(k_1, k_2)$ is a coalition at time $t$,

$$\begin{aligned} cost^t_{\pi(k)} &\overset{Lem.1}{=} \alpha_{\pi(k)} ccpa_{\pi(k_1,k_2)} \\ &\overset{Def.6}{=} \alpha_{\pi(k)} \frac{\sum_{j=k_1}^{k_2} mc_{\pi(j)}}{\sum_{j=k_1}^{k_2} \alpha_{\pi(j)}} \\ &= \left( \sum_{j=k_1}^{k_2} mc_{\pi(j)} \right) \frac{\alpha_{\pi(k)}}{\sum_{j=k_1}^{k_2} \alpha_{\pi(j)}}, \end{aligned}$$

which is similar to proportional cost sharing where the total cost (of all passengers) is distributed among all passengers proportionally to their alpha values.

The sum of the shared costs of all passengers in any coalition ("the shared cost of the coalition") at time $t$ equals the sum of the marginal costs of all passengers in the coalition ("the marginal cost of the coalition") at the same time since, for all times $k_1$, $k_2$ and $t$ and all arrival orders $\pi$ with $k_1 \le k_2 \le t$ such that $(k_1, k_2)$ is a coalition at time $t$,

$$\begin{aligned} \sum_{j=k_1}^{k_2} cost^t_{\pi(j)} &\overset{Lem.1}{=} ccpa_{\pi(k_1,k_2)} \sum_{j=k_1}^{k_2} \alpha_{\pi(j)} \\ &\overset{Def.6}{=} \frac{\sum_{j=k_1}^{k_2} mc_{\pi(j)}}{\sum_{j=k_1}^{k_2} \alpha_{\pi(j)}} \sum_{j=k_1}^{k_2} \alpha_{\pi(j)} \\ &= \sum_{j=k_1}^{k_2} mc_{\pi(j)}, \end{aligned}$$

which is similar to incremental cost sharing. where the shared cost of a passenger is its marginal cost. It also implies the budget-balance property since summing over all passengers in all coalitions is identical to summing over all passengers and the sum of the marginal costs of all passengers equals the total cost.

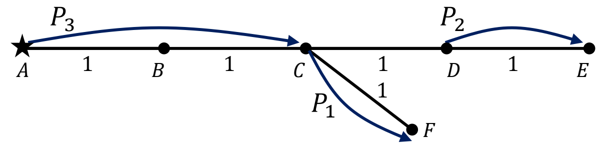



**Figure 2: DRT Example 2**

**Table 4: POCS:** $ccpa_{\pi(k_1,k_2)}$

| | $k_2 = 1$ $\pi(k_2) = P_1$ | $k_2 = 2$ $\pi(k_2) = P_2$ | $k_2 = 3$ $\pi(k_2) = P_3$ | $k_2 = 4$ $\pi(k_2) = P_4$ |
|---|---|---|---|---|
| $k_1 = 1$ $\pi(k_1) = P_1$ | 20 | 30 | 15 | 16 |
| $k_1 = 2$ $\pi(k_1) = P_2$ | | 40 | 13 1/3 | 15 |
| $k_1 = 3$ $\pi(k_1) = P_3$ | | | 0 | 6 2/3 |
| $k_1 = 4$ $\pi(k_1) = P_4$ | | | | 20 |

## 3.3 Illustration

Table 4 shows the coalition costs per alpha value for the DRT example in Section 2.2. The coalition costs per alpha value are used to calculate the shared costs, shown in Table 5. The shared costs, in turn, are used to calculate the shared costs per alpha value, shown in Table 6, by dividing the shared costs by the alpha values, shown in Table 1. For example, at time 4, Passengers $P_1$, $P_2$ and $P_3$ form a coalition (since their shared costs per alpha value are equal), and Passenger $P_4$ forms a coalition by itself. The sum of the marginal costs of the three passengers in the first coalition ("the total cost of all passengers in the coalition") is 120 and is distributed among all passengers in the coalition proportionally to their alpha values, namely 2, 2 and 4, respectively. Consequently, the shared cost of Passenger $P_3$ at time 4 and thus its fare is $cost^4_{\pi(3)} = 60$. Table 6 shows that the shared costs per alpha value in each row are monotonically non-decreasing from left to right, corresponding to the online-fairness property. Table 5 shows that the shared costs in each column are monotonically non-increasing from top to bottom (and consequently Table 6 shows that the shared costs per alpha value have the same property), corresponding to the immediate-response property. Table 5 also shows that the sum of the shared costs in each row equals the total cost at the corresponding time, corresponding to the budget-balance property.

## 3.4 Ex-Post Incentive Compatibility

We use the DRT example in Figure 2 to illustrate that POCS does not satisfy the ex-post incentive-compatibility property if the second condition (namely that the other passengers do not change whether they accept their fare quotes or drop out) is removed. There is one shuttle that can transport up to four passengers and starts at the star. The shuttle incurs an operating cost of 10 for each unit of distance traveled and needs to return to its initial location. There are three passengers. Passengers $P_1$ and $P_3$ accept all fare quotes, while Passenger $P_2$ accepts all fare quotes up to 60. Assume that the passengers arrive in order $P_1$, $P_2$ and $P_3$. First, Passenger $P_1$ arrives, receives a fare quote of 60 and

**Table 5: POCS:** $cost^t_{\pi(k)}$

| | $k=1$ $\pi(k)=P_1$ | $k=2$ $\pi(k)=P_2$ | $k=3$ $\pi(k)=P_3$ | $k=4$ $\pi(k)=P_4$ |
|---|---|---|---|---|
| $t=1$ | 40 | | | |
| $t=2$ | 40 | 80 | | |
| $t=3$ | 30 | 30 | 60 | |
| $t=4$ | 30 | 30 | 60 | 40 |

**Table 6: POCS:** $cost^t_{\pi(k)}/\alpha_{\pi(k)}$

| | $k=1$ $\pi(k)=P_1$ | $k=2$ $\pi(k)=P_2$ | $k=3$ $\pi(k)=P_3$ | $k=4$ $\pi(k)=P_4$ |
|---|---|---|---|---|
| $t=1$ | 20 | | | |
| $t=2$ | 20 | 40 | | |
| $t=3$ | 15 | 15 | 15 | |
| $t=4$ | 15 | 15 | 15 | 20 |

accepts it. Second, Passenger $P_2$ arrives, receives a fare quote of 50 and accepts it. Third, Passenger $P_3$ arrives, receives a fare quote of 50 and accepts it. In the end, Passengers $P_1$, $P_2$ and $P_3$ are serviced with fares of 25, 25 and 50, respectively. Now assume that Passenger $P_1$ delays its arrival, and the passengers arrive in order $P_2$, $P_3$ and $P_1$. First, Passenger $P_2$ arrives, receives a fare quote of 80 and drops out since the fare quote exceeds its fare limit of 60. Second, Passenger $P_3$ arrives, receives a fare quote of 40 and accepts it. Third, Passenger $P_1$ arrives, receives a fare quote of 20 and accepts it. In the end, Passengers $P_1$ and $P_3$ are serviced with fares of 20 and 40, respectively. Thus, Passenger $P_1$ managed to decrease both its fare quote and fare by delaying its arrival since this caused Passenger $P_2$ to drop out.

# 4. EXPERIMENTAL ANALYSIS

We have proved that POCS satisfies five properties that make DRT systems more attractive to both shuttle providers and passengers, provided that our assumptions are satisfied. For example, Definition 2 assumes that the total cost satisfies two properties that hold for the minimal operating cost, which is therefore the quantity that we have used so far for the total cost. Calculating the minimal operating cost is typically an NP-hard problem and thus time-consuming. However, DRT systems need to calculate the minimal operating cost every time a ride request is submitted, which would prevent them from operating in real-time. We thus present an experimental study with a transport simulation where the DRT system uses a heuristic to compute a low operating cost that is not guaranteed to be minimal [10]. In this case, the assumption in Definition 2 that the total cost is independent of the arrival order of passengers (which implies that the decisions of passengers to accept their fare quotes or drop out and thus also their fare quotes themselves do not depend on the arrival order of passengers) is not satisfied. This assumption is used (only) to prove that POCS satisfies the ex-post incentive-compatibility property. We thus investigate whether the best strategy of every passenger remains to arrive truthfully, for example because the likelihood of transport capacity still being available tends to decrease over time.

## 4.1 Transport Simulator

Our transport simulator first generates a given number of shuttles and passengers. Each shuttle is characterized by its capacity, start location, end location, operating time window and operating cost for each unit of distance traveled. Each passenger is characterized by its truthful arrival time, start location, end location, pick-up time window, drop-off time window and fare limit. The settings of our simulator are slightly more general than what we have used in the DRT examples because operating time windows of shuttles and pick-up and drop-off time windows of passengers are taken into account. The transport simulator then simulates each passenger. Once a passenger is assigned to a shuttle, it is never re-assigned to a different shuttle, which makes it possible to calculate the marginal cost of a passenger as the lowest operating cost increase of adding the passenger to any shuttle, but is also a reason why the total cost (which equals the sum of the operating costs of all shuttles) is not guaranteed to be equal to the minimal operating cost or to be independent of the arrival order of the passengers. When a new passenger submits a ride request, the transport simulator requests from each shuttle the operating cost increase from adding the passenger to all passengers previously assigned to it, selects a shuttle with the lowest operating cost increase and then uses POCS to calculate a fare quote for the passenger under the assumption that the passenger is assigned to the selected shuttle. If the fare limit of the passenger is lower than this fare quote, then the passenger drops out, and the transport simulator does not service it. Otherwise, the passenger accepts the fare quote, and the transport simulator adds it to all passengers previously assigned to the selected shuttle and then updates the shared costs of all passengers assigned to the shuttles.

Each shuttle has to calculate its route, schedule and operating cost increase (or, equivalently, operating cost) when adding a new passenger to all passengers previously assigned to it. The shuttle maintains an itinerary for all passengers assigned to it - in the form of a sequence of locations, namely its start location, its end location and the start and end locations of all passengers assigned to it. It calculates its travel distance as the shortest travel distance needed to visit all locations in the order given in its itinerary, and it calculates its operating cost as the product of its travel distance and its operating cost for each unit of distance traveled. Determining an itinerary for the new passenger and all passengers previously assigned to it that minimizes its operating cost is time-consuming. The shuttle therefore uses a non-optimal scheduling method [17, 11], which is another reason why the total cost is not guaranteed to equal the minimal operating cost and not guaranteed to be independent of the arrival order of passengers. In the construction phase of the scheduling method, the shuttle uses a cheapest-insertion method to construct a (feasible) itinerary by inserting the start and end locations of the new passenger into the cached itinerary for the passengers previously assigned to it. In the subsequent improvement phase of the scheduling method, the shuttle uses tabu search [7, 13, 5, 6], a form of hill climbing, to improve the itinerary from the construction phase.

## 4.2 Experiment 1

In Experiment 1, we demonstrate that passengers have an incentive to arrive truthfully since their fare quotes and fares tend to increase as their arrival times increase. Thus, it is more likely that they accept their fare quotes and are serviced for low fares if they arrive as early as possible. We perform 10,000 simulations with the transport simulator in a grid city of size $11 \times 11$ (that is, with 121 locations) and

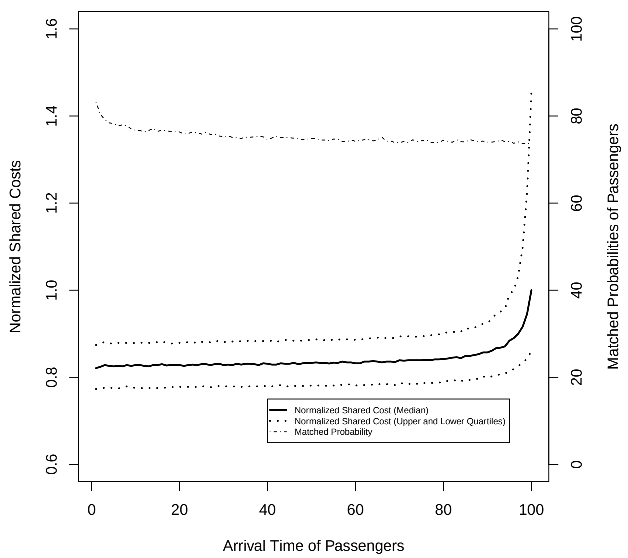



**Figure 3: Results of Experiment 1**

report average results. There are 25 shuttles that can each transport up to 10 passengers and operate the same hours from dawn (time 101) to dusk (time 1440). We assume that passengers submit their requests before dawn (the departure time of the shuttles) because otherwise the marginal costs depend on their arrival times. We also assume that shuttles have sufficient time to service all passengers before dusk. The shuttles start at a depot in the center of the city. Each shuttle incurs an operating cost of 1 for each unit of distance traveled and needs to return to its initial location at dusk. There are 100 passengers that all arrive truthfully one at a time (that is, their arrival times range from time 1 to time 100). The start location of 20 percent of the passengers is the depot. The start locations of the other passengers and the end locations of all passengers are randomly selected from all locations with uniform probability. The pick-up and drop-off time windows are identical for each passenger but might be different from passenger to passenger. Their lower bounds are dawn, and the differences between their upper and lower bounds are randomly selected from being 2.5 to 3.0 times higher than their alpha values (that is, the shortest point-to-point travel distance from their start location to their end location). Thus, passengers do not have tight schedules, resulting in low fare quotes. The fare limits of passengers are randomly selected from being 1.5 to 3.0 times higher than their alpha values. Thus, passengers have high fare limits. For both of these reasons, the fare quotes often do not exceed the fare limits. Many passengers therefore accept their fare quotes and are serviced.

Figure 3 shows the probability that passengers accept their fare quotes ("Matched Probabilities of Passengers") as a function of their arrival times $k$, that is, the percentage of simulations with $cost^k_{\pi(k)} \leq w_{\pi(k)}$. The probability that passengers accept their fare quotes is around 75 percent. It decreases as their arrival times increase (since their fare quotes tend to increase as their arrival times increase) but only very slowly. Figure 3 also shows the fares per alpha value of all passengers that accepted their fare quotes ("Normalized Shared Costs") as a function of their arrival times $k$, that is, $cost^{100}_{\pi(k)}$ averaged over all simulations with $cost^k_{\pi(k)} \leq w_{\pi(k)}$. The fares per alpha value of passengers increase as their arrival times increase (as suggested by the online fairness property) but only very slowly. The only exception is the sharp increase for arrival times close to 100 since passengers that arrive then can no longer share their costs with a high number of passengers that arrive after them.

## 4.3 Experiment 2

The definition of ex-post incentive compatibility states that the best strategy of every passenger is to arrive truthfully, provided that all other passengers arrive truthfully as well and do not change whether they accept or decline their fare quotes, two assumptions that are not guaranteed to be satisfied in practice. We have already shown in Section 3.4 that POCS does not satisfy the ex-post incentive-compatibility property if the second condition is removed. In Experiment 2, we therefore evaluate how likely it is that passengers can decrease their fares by delaying their arrivals if the second condition is removed. Experiment 2 is similar to Experiment 1, except that we distinguish four scenarios with different flexibilities of shuttles and passengers and use experimental parameters that decrease the scale of the experiment since each simulation is now more time-consuming. We perform 1,000 simulations with the transport simulator in a grid city of size $5 \times 5$ and report average results. Each simulation consists of at most 45 runs in addition to a run where Passengers $P_1 \ldots P_{10}$ arrive truthfully in order $P_1 \ldots P_{10}$, namely runs where all passengers arrive truthfully except that Passenger $P_i$ delays its arrival and arrives only immediately after Passenger $P_j$ for all $i$ and $j$ with $1 \leq i < j \leq 10$ where Passenger $P_i$ accepts its fare quote when all passengers arrive truthfully. There are either 2 or 10 shuttles (for two scenarios) that can each transport up to 3 passengers, operate the same hours from dawn to dusk and start at a depot in the center of the city. Each shuttle incurs an operating cost of 1 for each unit of distance traveled and needs to return to its initial location at dusk. There are 10 passengers that arrive one at a time (that is, their arrival times range from time 1 to time 10) before the shuttles start to service them. The start and end locations of all passengers are randomly selected from all locations with uniform probability. The pick-up and drop-off time windows are identical for each passenger but might be different from passenger to passenger. Their lower bounds are dawn, and the differences between their upper and lower bounds are either 3.0 or 4.0 times (for two scenarios) higher than their alpha values. The fare limits of passengers are 3.0 times higher than their alpha values.

Table 7 shows, for each scenario, both the number of runs and the probabilities that passengers who delay their arrivals improve (since their fares decrease), do not change (since their fares remain unchanged) or worsen (since either their fare quotes increase sufficiently for them to drop out or - in case they do not drop out - their fares increase) their situations. Experiment 2 demonstrates that passengers have an incentive to arrive truthfully since, in all scenarios, the probability that passengers who delay their arrivals improve their situations is lower than 20 percent while the probability that they worsen their situation is higher than 50 percent. Exper-

**Table 7: Results of Experiment 2**

| Scenario | Number of Shuttles | Time Window | Number of Runs | Situation Improves | No Change | Situation Worsens | |
|---|---|---|---|---|---|---|---|
| | | | | | | Not Dropping Out | Dropping Out |
| 1 | 2 | 3.0 | 32,808 | 11% | 32% | 24% | 33% |
| 2 | 2 | 4.0 | 37,259 | 15% | 31% | 39% | 15% |
| 3 | 10 | 3.0 | 36,955 | 16% | 31% | 51% | 2% |
| 4 | 10 | 4.0 | 37,990 | 17% | 29% | 51% | 3% |

iment 2 does not measure one advantage of passengers who delay their arrivals, namely the situation when passengers originally dropped out since their fare quotes exceeded their fare limits and by delaying their arrivals improve their fare quotes so they no longer drop out. Also, Experiment 2 assumes that passengers delay their arrivals randomly (rather than strategically) due to missing knowledge of future arrivals of passengers. The probability that the situation for passengers who delay their arrivals worsens is zero if passengers are able to delay their arrivals strategically since they can always decide to arrive truthfully instead, in which case their situations do not change. We thus expect the probability that their situations improve to increase.

## 5. CONCLUSIONS

In this paper, we determined properties of cost-sharing mechanisms that we believe make demand-responsive transport systems attractive to both shuttles and passengers, namely online fairness, immediate response, individual rationality, budget balance and ex-post incentive compatibility. We then proposed a novel cost-sharing mechanism, called Proportional Online Cost Sharing (POCS), that has these properties. Overall, POCS is a first step towards addressing some of the problems raised by the missing knowledge of future arrivals of passengers, which differentiates our research from previous research [3, 15, 8, 1]. However, some issues remain to be addressed by more advanced online cost-sharing mechanisms, including integrating more complex models of passengers, shuttles and transport environments. Our current simplifying assumptions include, for example, that the availability of shuttles does not change unexpectedly, that all passengers arrive before the shuttles start to service passengers, that fares depend only on the ride requests and no other considerations (for example, that DRT systems do not face competition), that all passengers evaluate their trips uniformly according to the criteria quantified by the alpha values (for example, that all passengers consider travel time to be equally important), that DRT systems provide fare quotes to passengers without predicting future arrivals of passengers (for example, that DRT systems service hard-to-accommodate passengers even though these passengers increase the shared costs of subsequent passengers and might make subsequent passengers drop out), that passengers try to decrease their fares only by delaying their arrivals (rather than, for example, by colluding with other passengers or entering fake ride requests under false names) and that passengers honor their commitments (for example, that passengers do not change ride requests, cancel them or show up late).

## 6. REFERENCES

[1] R. Anderson and A. Claus. Cost allocation in transportation systems. *Southern Economic Journal*, pages 793–803, 1976.

[2] J. Brenner and G. Schäfer. Online cooperative cost sharing. In T. Calamoneri and J. Diaz, editors, *Algorithms and Complexity*, volume 6078, pages 252–263. Springer, 2010.

[3] M. Frisk, M. Göthe-Lundgren, K. Jörnsten, and M. Rönnqvist. Cost allocation in collaborative forest transportation. *European Journal of Operational Research*, 205(2):448–458, 2010.

[4] M. Furuhata, L. Cohen, and S. Koenig. Using auctions to allocate transportation requests for demand responsive transit systems. Technical Report 09-19, Metrans Transportation Center, University of Southern California, Los Angeles, USA, 2013.

[5] M. Gendreau. An introduction to tabu search. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research and Management Science*, pages 37–54. Springer, 2003.

[6] M. Gendreau and J. Potvin. Tabu search. In E. Burke and G. Kendall, editors, *Search Methodologies*, pages 165–186. Springer, 2005.

[7] F. Glover, E. Taillard, and D. de Werra. A user's guide to tabu search. *Annals of Operations Research*, 41(1):1–28, 1993.

[8] J. Lemaire. An application of game theory: Cost allocation. *Astin Bulletin*, 14(1):61–81, 1984.

[9] H. Moulin. Incremental cost sharing: Characterization by coalition strategy-proofness. *Social Choice and Welfare*, 16(2):279–320, 1999.

[10] H. Moulin. Cost sharing in networks: Some open questions. *International Game Theory Review*, 15(2):1–10, 2013.

[11] W. Nanry and J. Barnes. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B*, 34(2):107–121, 2000.

[12] D. Parkes. Online mechanisms. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 16, pages 411–439. Cambridge University Press, 2007.

[13] C. Reeves. *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell, 1993.

[14] Y. Sprumont. Ordinal cost sharing. *Journal of Economic Theory*, 81(1):126–162, 1998.

[15] F. Tohmé and T. Sandholm. Coalition formation processes with belief revision among bounded-rational self-interested agents. *Journal of Logic and Computation*, 9(6):793–815, 1999.

[16] Y. Wang and D. Zhu. Ordinal proportional cost sharing. *Journal of Mathematical Economics*, 37(3):215–230, 2002.

[17] Z. Xiang, C. Chu, and H. Chen. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European Journal of Operational Research*, 174(2):1117–1139, 2006.

# Toward Equitable Vehicle-based Intersection Control in Transportation Networks

Emmanuel Katende Dinanga *
Centro de Tecnologia, Universidade Federal de Santa Maria
Av. Roraima 1.000 - Cidade Universitária - CEP 97.105-900
Santa Maria - RS, Brazil
emmanuel@inf.ufsm.br

Marcia Pasin†
Centro de Tecnologia, Universidade Federal de Santa Maria
Av. Roraima 1.000 - Cidade Universitária - CEP 97.105-900
Santa Maria - RS, Brazil
marcia@inf.ufsm.br

## ABSTRACT

Efficient intersection control is an interesting problem in traffic management, and may collaborate to reduce traffic jams as well travel times. New technologies, such as Vehicular Ad hoc NETworks (VANETs) and ubiquitous computing, may collaborate to the implementation of new policies to intersection control, thus providing flexibility and performance to transportation networks. While these technologies are not widely available, new policies to intersection control can be intensively evaluated in simulation environments. In this paper, we evaluate different intersection control policies and different scenarios using as support SUMO, a transportation network simulator in the context of multiagent systems (MAS). In the evaluation, we concern in equitability which measures the fairness to attend a request from a vehicle to pass a given intersection. Our simulation results indicate that different policies are suitable to different scenarios leading us to believe that adaptive policies must be proposed.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

## General Terms

Experimentation

## Keywords

Intersection control, Simulation, Multiagent system, Transportation networks.

*Undergraduate student. Curso de Ciência da Computação.
†Associate Professor. Departamento de Linguagens e Sistemas de Comutação.

## 1. INTRODUCTION

With the increasing number of vehicles circulating in urban areas, and the consequent increase in demand, the development of services supported by information and communication technologies (ICTs) to improve traffic management and the provision of urban mobility are indispensable. In this scenario, new technologies such as VANETs (Vehicular Ad hoc NETworks), ubiquitous computing and cloud computing allow adequate infrastructure for such services. In the future, vehicles will be able to share information in transportation networks, and will be able to collaborate to reduce traffic jams, travel times, accidents and vehicle emissions.

Intersection control represents a major challenge in traffic management, and it means to decide which vehicle should pass an intersection and which vehicle should wait. In real traffic systems, intersection control is solved by traffic lights, or using priority signs, or by *the priority to the right* rule, when the intersection is not signalized. Traffic lights traditionally control vehicles' flow using signal-timing plan with unique set of timing parameters. The large majority of traffic lights cannot apart in presence of changes in traffic conditions and it can result in inefficient service.

With the availability of VANETs infrastructure and services, traffic lights would be eliminated. Vehicles will be provided with GPS devices and vehicle to vehicle (V2V), and vehicle to infrastructure (V2I) communication, installed and operational. Road intersection control would be performed by the vehicles themselves, modeled as autonomous agents. In this scenario, each autonomous agent independently obeys its own behavior and interacts each other and/or with the infrastructure allowing the decision-making process.

Thus, adaptive solutions can be applied. Dynamic solutions adapt behavior according to the traffic flow and can be centered at the vehicle flow, and, alternatively at the vehicles themselves. Semaphores based on adaptive flows have been established in some Brazilian cities (e.g. Curitiba, Porto Alegre, Belo Horizonte, and Fortaleza). They are calibrated using information provided by the vehicles' flow and aim to eventually reduce congestion and travel times. The mode of operation is simple: sensors are installed on the tracks and capture the presence of vehicles. This information is used as input to calculate the proper *split* and *cycle length* in signal-timing plans. This solution would avoid, for example, the exposure of green light for a prolonged period in a road with few vehicles, if the traffic is heavy at the concurrent flow. However, it does not eliminate the need of having a physical device installed and in operation.

Installation and maintenance of traffic lights is considerable expenses in Brazilian cities and in many world wide cities. For instance, in Porto Alegre, there are more than 1,007 signalized intersections. The cost of installing each semaphore is between $5,000 -$7,000 (Source: EPTC March 2011). In São Paulo, there are more than 4,800 signalized intersections (Source: CET 2013). In Fortaleza, there are 656 traffic lights and the mensal cost to maintenance is about $160,000 (Source: AMC June 2013). According to Ferreira *et al.* [6] maintenance of traffic lights is considerable expenses in the budget of cities. Thus, eliminating traffic lights can result in budget savings.

In a futuristic scenario, with the deployment of VANETs and the concept of autonomous vehicles, traffic lights would be completely eliminated. Intersection control will be undertaken by vehicles themselves. Indeed, the ability to implement policies to intersection control with VANETs support contrasts with the traditional signal-timing plans, which uses a mathematical model to describe the traffic flow. Therefore, new policies to deal with intersection control, beyond the traditional signal-timing plans adopted by traffic lights, must be proposed and evaluated before the availability of new technologies. For instance, policies to deal with CPU scheduling, such as FIFO (first in first out) and SJF (shortest job first), would be used to control the vehicle passage through intersections, with the support of V2V and V2I communication.

Since VANETs technology is not yet widely available, computer simulation gives a way to evaluate possibilities before being implemented them in practice. Thus, in this paper, we evaluate different intersection control policies using simulation supported by multiagent systems (MAS), and V2I communication. Each vehicle is represented as an autonomous agent that follows a behavior independently and interacts with other agents and/or infrastructure for decision-making. Bazzan [1] and Chen *et al* [3] emphasize the benefits of using MAS to model and to evaluate solutions target to transportation systems. Experiments were conducted in SUMO [2]. In the evaluation, we concern in equitability which measures the fairness to attend a request from a vehicle to pass through an intersection. Our simulation results indicate that different policies are suitable to different scenarios leading us to believe that adaptive policies must be proposed.

The paper is organized as follows. Related works are described in section 2. Algorithms and metrics we used in experiments were described in Section 3. Experimental evaluation results are presented in Section 4. Finally, conclusions and future works are presented in Section 5.

## 2. RELATED WORKS

The idea of removing traffic lights or at least to improve its use is not new. In the following we discuss some existing research projects.

Krajzewicz *et al.* [7] focused on efficient flow-sensitive traffic lights. With the support of SUMO, Krajzewicz *et al.* compares the size of different of vehicle queues' to decide about which vehicle will pass an intersection first. The decision is made using the support of V2I communication. Vehicles in the larger queue have the higher priority to cross the intersection. Intersection control is performed by a physical device implemented by the infrastructure (i.e. not by the vehicles themselves). V2V communication is not taken into account.

Vehicle centered-solutions would be one step further, and would use some mechanism to promote not only efficient traffic flow, but also fairness to attend user service. However, vehicular communication technology must be widely available. Dresner & Stone [4] describe a reservation scheme where the vehicle should allocate a slot, in a central, concerning space and time to cross an intersection of two roads. According to the experiments presented in this article, this technique would be more efficient in terms of throughput in comparison with the traditional semaphore. However, if a vehicle cannot book a slot necessary to cross of the intersection, it can suffer indefinite hold. This drawback was fixed in [5]. Another problem is the existence of a central to apply the intersection control policy. If the system fails, the service becomes unavailable. An extension of the work of Dresner & Stone for the context of multiple intersections was conducted by Vasirani & Ossowski [9]. The idea is to provide an adequate service to the public, but still without collaboration among vehicles.

Finally, in Ferreira *et al.* [6], through the support of V2V communication and AVL (automatic vehicle location), the nearest vehicle to an intersection is elected to coordinate the passage of vehicles at a particular intersection. When the driver finally passes the intersection, a new vehicle is chosen to manage the intersection. However, given that two vehicles $v_i$ and $v_j$ may be placed in distinct pathways $S$ and $W$, but share the same distance $d$ with respect to the intersection, a guarantee of election only one coordinator needs to be imposed. Furthermore, fairness to attending user service is not taken into account.

We may conclude that there is a need of research works to evaluate more effectively intersection control and to explore more broadly these mechanisms. The response to the request of vehicles passing through an intersection must be performed efficiently (through a solution that delivers traffic flow) and in the direction to minimize the waiting time of each vehicle individually. By minimizing the waiting time, we mean that the policy applied to intersection control must look for equitability or fairness. Vehicles in different queues should not waiting so long to pass through an intersection. In addition, starvation must be avoided. This work is a step toward this direction.

In general, new mechanisms to intersection control need to be proposed and should be analyzed extensively before putting them into practice and before VANETS technology would be widely available. Additionally, it would be interesting the use of policies preferably focused on vehicular communication to enable the exclusion (physical) traffic lights. This will be the focus on our future work.

## 3. MODELING THE PROBLEM

### 3.1 Applying the agent model to transportation networks

Autonomous agents is a convenient abstraction to model transportation networks. Vehicles are described such as autonomous units with independent behavior. In our scenario, a group of agents (vehicles) follows a policy to pass of an intersection point.

Each vehicle $v_i$ (to $i \in \mathbb{N}$) is uniquely identified (in practice, the Vehicle Identification Number (VIN) can be used) and belongs to an unique queue or segment called $S$ or $W$. Queue $S$ is placed in the route $SN$ and queue $W$ is placed in the route $WE$. All the vehicles in $S$ move from South to North, while all the vehicles in $W$ move from West to East. The queues share a critical section (intersection point $P$), where vehicles must pass to reach an ultimate goal. Only one vehicle may pass the critical section at a time. Who decides what is this vehicle is the **intersection control policy**. Figure 1 depicts this scenario.

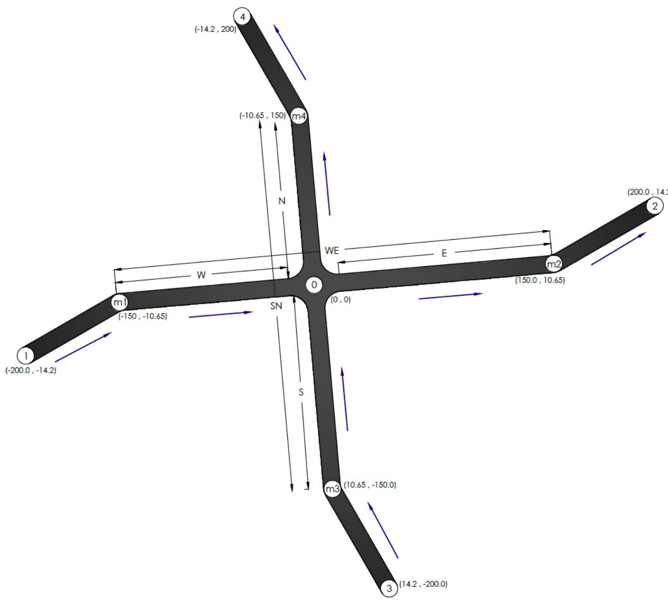



Figure 1: Scenario with routes $SN$ and $WE$ and segments $S$ and $W$

Also, in Figure 1, there is the information used to code this intersection and routes in SUMO, the transportation network simulator in the context of MAS we used to implement and compare policies.

In our approach, basically, each vehicle $v_i$ from $S$ or $W$ comes in contact with the infrastructure, through the emission of a message $m_i$ to order a ticket to allow passing through the intersection, according to the policy in question. The infrastructure applies the policy and decides the order in which the vehicle must pass the intersection, and informs it to the vehicle through a reply message $m_i - 1$. The infrastructure can be implemented in distributed or in a centralized fashion, and will be focused in our future work. In practice, V2I communication could be supported by the IEEE 802.11 p protocols (Wi-Fi) or by GSM/GPRS and 3G/4G (i.e. mobile phone networks).

## 3.2 Intersection control policies

The scope of this work, we implemented five policies to intersection control, including: ($i$) the right of way, ($ii$) signal-timing plan, ($iii$) the largest queue always, ($iv$) the largest queue first, and ($v$) at least $k$ vehicles each time. These policies are described more in detail in the following.

### 3.2.1 *The right of way*

The default policy, *the right of way*, results from the settings taken by SUMO to generate the simulation. This policy is based on assigning the highest priority for the passage of vehicles through intersections using the right of way policy. Considering two road segments $S$ and $W$ that meet at an intersection point $P$, and suppose that the highest priority of passing vehicles by $P$ is assigned to $S$, vehicles on $W$ only pass through $P$ when no vehicles are queuing on $S$.

### 3.2.2 *Signal-timing plan*

*The signal-timing plan* is the policy applied by traditional traffic lights to deal with intersection control. It is based on the scheduling traffic signal phases at intervals given by phases.

The major drawback of signal-timing plan applied in large majority of traffic lights is that it cannot apart in presence of changes in traffic conditions, and it can result in inefficient service. For instance, it cannot avoid presenting the green signal for a long period of time even if there is only one vehicle or a few vehicles in a queue.

### 3.2.3 *The largest queue always*

*The largest queue always policy* consists of giving the higher priority to the passage through the intersection to the segment with larger queue of vehicles outside the critical section.

Considering two road segments $S$ and $W$ that meet at an intersection point $P$, the algorithm of longest line always starts capturing all vehicles outside the critical section in $S$ and $W$, calculating the number of vehicles on each track segment, and comparing the two values. If $S$ is the largest queue, vehicles on queue $W$ need wait, until the last vehicle in queue $S$ passes through $P$. Next, the lengths of the queues are compared again to decide who will pass through $P$. The same process is repeated until the end of the simulation.

The drawback of this policy is that it can suffer from starvation in case of a queue is typically shortest than the other, even if new vehicles are continuously added on queues.

### 3.2.4 *The largest queue first*

The policy defined by *the largest queue first* is similar to *the largest queue always*, except that it does not only give priority of passing through the intersection to the track segment with the line of vehicles outside the critical section. However, it lets the lower queue of the other track segment to cross the intersection, before returning to compare the two queues lengths' again.

Considering two road segments $S$ and $W$ that meet at an intersection point $P$, *the largest queue first* starts capturing all vehicles out of the critical section in $S$ and $W$, calculating the number of vehicles on each track segment, and comparing the two values. If $S$ is the largest queue, vehicles on queue $W$ need wait, until the last vehicle in queue $S$ passes through $P$. Then it passes the entire row in $W$ before comparing again the next queues in both segments. The same process is repeated until the simulation ends. Contrasting with *the largest queue always policy*, in *the largest queue first* starvation does not take place.

### 3.2.5 *At least $k$ vehicles each time*

The policy *at least $k$ vehicles each time* constitutes successive passage of vehicles of each road, since the number of vehicles on the road that has the slot to spend is greater than or equal to $k$, $k$ being an informed integer.

Considering two road segments $S$ and $W$ that meet at an intersection point $P$, and $S$ the track segment chosen to start the time. The policy *at least k vehicles each time* starts capturing all vehicles out of the critical section in $S$, and calculating the number vehicles on that queue. If this is greater than or equal to $k$, the vehicles on queue in $W$ need to stop until the last vehicle in the queue $S$ passes through the intersection. Otherwise, it turns passes to $W$. The same process is repeated until the end of the simulation.

## 3.3 Metrics

To compare and evaluate policies described in item 3.2 regarding equitability, we used a sort of specific metrics. Equitability measures the fairness to attend a request from a vehicle to pass through an intersection. In the following, we define these metrics.

**Definition 1: state of traffic flow**. A **state of traffic flow** or **state** $E_i$, for short, is the behavior the traffic flow, described from the period, the preference from a route with respect to another (i.e. priority), and maximum speed allowed in a given route (MaxSpeed). All these values are configured in SUMO.

**Definition 2: scenario**. A **scenario** $C_i$ is the result of applying one of the algorithms described in item 3.2 in on the defined traffic states $E_i$.

**Definition 3: rate of change of vehicles**. Considering a scenario $C_i$, with an intersection point $P$, and two track segments $S$ and $W$. The **rate of change of vehicles** in the range of $k$ steps, represented by $T_k$, is the ratio of the passage of vehicles originally in $X = \{S \text{ or } W\}$ by $P$ in the range of $k$ steps, defined by the following formula:

$$T_X k = \frac{\sum v_k}{\sum v}$$

where:

- $\sum v_k \in [0, \mathbb{N}]$ and $\sum v \in (0, \mathbb{N}]$, where $\mathbb{N}$ is the total of vehicles in a simulation.
- $k \in (0, N_s)$, where $N_s$ is the total number of *steps* in a simulation.
- $v_k$ represents a vehicle from $S$ or $W$ that passed through the intersection $P$ at the step $k$.
- $v$ represents a vehicle from $S$ or $W$ that still does not cross the intersection $P$.

Note that if $T_S k > T_W k$ at *step k*, then there were more vehicles from $S$ than in $W$ that pass through $P$ when the simulation reaches the *step k*.

**Definition 4: total rate of change**. Considering a scenario $C_i$, with an intersection point $P$, and the two track segments $S$ and $W$, the **total rate of change** of $N$ vehicles in the simulation, represented by $\overline{X}_N$ is the sum of the rates of variations in $N$. The total rate of change of a segment $X$ is defined by the following formula:

$$\overline{X}_X = \sum T_X$$

If $\overline{X}_S > \overline{X}_W$, then there were more vehicles $S$ that have passed through $P$ than in $W$.

**Definition 5: difference of total variation rates**. Considering a scenario $C_i$ with two road segments $S$ and $W$ that meet at in an intersection point $P$. The **difference of total variation rates** of $S$ and $W$ in $C_i$, represented by $d\overline{X}$, is the magnitude of the difference between $\overline{X}_S$ and $\overline{X}_W$, and is expressed by the following formula:

$$d\overline{X} = |\overline{X}_S - \overline{X}_W|$$

**Definition 6: distribution of the traffic flow concerning two road segments**. The **distribution of the traffic flow concerning two road segments** $S$ and $W$ crossing at an intersection point $P$ in scenario $C_k$, compares the proportions of vehicle crossings of $S$ and $W$ in $P$. Considering two scenarios $C_i$ and $C_j$ and the same $E$, and let $d\overline{X}_i$ and $d\overline{X}_j$, their difference in total charges in $C_i$ and $C_j$:

- if $d\overline{X}_i > d\overline{X}_j$, we say that the flow distribution in $C_i$ is better than $C_j$. In other words, $C_i$ is more distributed than $C_j$,
- if $d\overline{X}_i$ is tending to zero, we may say that the distribution flow in $C_i$ tends to be equitable, or $C_i$ tends to be equitable distributed with respect to the flow.

Finally, a policy $a$ is considered more effective than another $b$ based on a state of traffic flow $E_i$ if and only if the scenario $C_i$ generated by $a$ applied to $E_i$ is more equitable distributed than $C_j$ generated by $b$ applied to the same $E_i$.

# 4. EXPERIMENTAL EVALUATION

## 4.1 Configuration of traffic flow

To conduct experiments, we configured three different states of traffic flow, $E_{1-3}$, which are summarized on Table 1 and Figure 2. Each state of traffic flow occurs in a time interval of 14,400 steps. In SUMO, it represents 4 hours since a time step is, by default, one second. We believe this time interval is satisfactory to evaluate policies to and decide which algorithm implements the most efficient policy, since 4 hours represent a half-journey.

In the experiments, we used three parameters to configure traffic states: priority, period and *MaxSpeed*. If the priority of a road is higher that the priority of another road, that means if there are two vehicles on a intersection, the vehicle on the road with the highest priority goes first.

Period describes the traffic flow in terms of dense and rarefied. If a road has a shorter period than otherwise, it means that the road is denser than another road. With regard to the maximum speed, the value used is 16.7 m/s. In addition, the time interval used in *signal-timing plan* was 20$s$ to the *green* phase, 0$s$ to the *yellow* phase and 20$s$ to the *red* phase.

| **States** | **Priority** | **Period** | **MaxSpeed** |
|---|---|---|---|
| $E_1$ | SN > WE | SN = WE | SN = WE |
| $E_2$ | SN > WE | SN < WE | SN = WE |
| $E_3$ | SN > WE | SN > WE | SN = WE |

Table 1: States simulated in the experimental evaluation

In Table 1, the column *States* identify each traffic state. The column *Priority* displays the comparison of priority

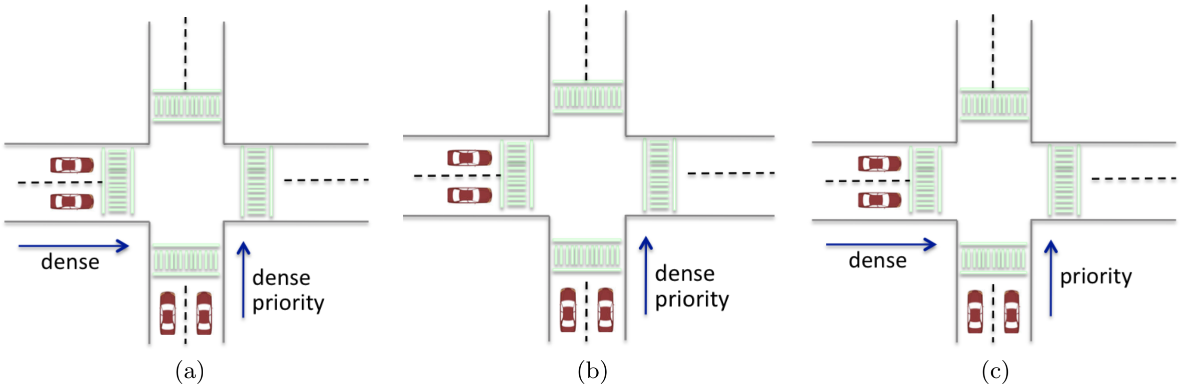



Figure 2: Configuration to states (a) $E_1$, (b) $E_2$ and (c) $E_3$

in routes $SN$ and $WE$, and column MaxSpeed display values used as maximum speed in the two routes. Remember that $SN$ represents the entire segment of road regarding an intersection $P$ from South to North, while $WE$ represents the entire segment of road regarding an intersection $P$ from West to East.

More specifically, as seen in Table 1, the state of traffic flow $E_1$ is defined as the state where the route $SN$ has the high priority in the passage of vehicles through the intersection $P$ with respect to via $WE$. The traffic flow is dense in the two ways, i.e. vehicles access the two routes with the same frequency, and are subject to the same maximum speed.

The state of traffic flow in $E_2$ and $E_3$ is the same used in $E_1$. However, in $E_2$ the traffic flow is dense only in $SN$. The traffic flow in $SN$ is ten times higher than in via $WE$. But all vehicles are subject to the same maximum.

The state of traffic flow $E_3$ has a back-flow to the $E_2$. The path $SN$ continues to have priority in the passage of vehicles through the intersection with respect to via $WE$, and the maximum speed achievable remains the same in both pathways. However, the traffic flow is dense only on via $WE$, i.e. vehicles access route $WE$ at a rate ten times higher than vehicles in via $SN$.

Given states $E_{1-3}$ and algorithms/policies described in Section 3, we combined them to obtained different group of scenarios. The scenarios generated are classified into three **groups of scenarios**, based on states of traffic flow in which the algorithms were applied. Basically, a scenario $C_i$ is a combination of a $E_i$ and a given policy.

## 4.2 Evaluating the results

To decide which policy is appropriate to a given scenario, we conduct experiments using SUMO and the previous given scenarios and values. The objective of the comparison is to find which is the most distributed scenario of each group in order to decide what the best algorithm that distributes the passage of vehicles through the intersection for each scenario group. The target variable used in the experimentation is the difference of total variation rates $d\overline{X}$ in each scenario.

Considering a scenario $C_i$ with two road segments $S$ and $W$ that meet at an intersection point $P$, and $T_S$ is the rate of change of the vehicles in segment $S$ in step $k$ and $T_W$ is the rate of change of vehicles at $W$ in step $k$, we may have that:

- $C_i$ is viewed through a graph, such as in Figure 3, that has two types of lines: those that represent rates of variation $T_w$ per step unit, and those that represent rates of variation per $T_s$ by step unit.
- $T_S$ indicates the percentage of vehicles in $S$ that passed by $P$ in step $k$ and $T_W$ indicates the percentage of vehicles in $W$ that passed by $P$ in step $k$.
- When $T_S$ is 0, it indicates that there is no vehicle in $S$ that have passed through $P$ in step $k$ and when $T_W$ is 0, it indicates that there is no vehicle in $W$ that have passed through $P$ in the step $k$.
- When $T_S$ is maximum, this indicates that all vehicles in $S$ have passed through $P$ in step $k$ and when $T_W$ is maximum, this indicates that all vehicles in $W$ have passed for $P$ in step $k$.

More specifically, in Figure 3, the scenario shown results from the application of *signal-timing plan policy* to traffic flow state $E_i$. Between 0 and 20 steps, for instance, $T_S$ and $T_W$ are equal to 0. Therefore, no vehicle from $S$ or $W$ has passed through $P$ in that interval. In addition, in any step in $T_S$ or $T_W$ is maximal. Therefore, there was never happened a situation in which all vehicles placed in $S$ or in $W$ passing through $P$.

Another example of a scenario is given in Figure 4, where the scenario results from the application of *the right of way policy* to the state of traffic flow $E_2$. Again, between 0 and 20 steps, for instance, $T_S$ and $T_W$ are equal to 0. Therefore, no vehicle from $S$ or $W$ has passed through $P$ in that interval. In addition, in step in $T_S$ or $T_W$ are maximal. Therefore, there was happened a situation in which all vehicles in $S$ or in $W$ passing through $P$.

At the total, 15 graphs were generated. Due lack of space in this document, we will not present all the graphs here. Our discussion will be based on tables, which will be addressed below. Each table is associated with a previous described scenario, $E_{1-3}$, grouping the three different categories. In the following, we discuss these group of scenarios.

### 4.2.1 *Scenario Group* $G_1$

Table 2 summarizes the results obtained by the application of state $E_1$ to the implemented policies within the range of 14,400 steps in SUMO.

Regarding the obtained results, one can observe that the policy *the largest queue first* is the most widely distributed

| Policy | $X_s$ | $X_w$ | $d\overline{X}$ |
|---|---|---|---|
| The right of way | 1,626.13 | 41.62 | 1,584.51 |
| Signal-timing plan | 463.71 | 196.59 | 267.12 |
| The largest queue always | 400.18 | 301.56 | 98.62 |
| The largest queue first | 277.21 | 334.54 | 57.32 |
| At least $k$ vehicles each time | 395.33 | 282.58 | 112.75 |

Table 2: Traffic flow distribution $d\overline{X}$ concerning two road segments obtained by the application of state $E_1$ to the implemented algorithms within the range of $14,400$ steps in SUMO

of all with a $d\overline{X} = 57.32$. Remember the smallest $d\overline{X}$, the most effective in terms of equitability. The worst policy is *the right of way* with $d\overline{X} = 1{,}584.51$.

Thus, we can conclude that the policy implemented in *the largest queue first* proved to be the best choice to control the passage of vehicles through the intersection between two lines with the same flow frequency and maximum speed, a priority which is higher than the other, so as defined by state of traffic flow $E_1$.

In the following, we have the policy *the largest queue always* similar to *at least k vehicles each time*, which proved to be the second and the third most suitable policies for the control of such traffic. The *signal-timing plan* is the forth choice and *the way of right* policy is the least suitable for such transit.

### 4.2.2 Scenario Group $G_2$

Table 3 summarizes the results obtained by the application of state $E_2$ to the implemented algorithms within the range of $14,400$ steps in SUMO.

| Policy | $X_s$ | $X_w$ | $d\overline{X}$ |
|---|---|---|---|
| The right of way | 1,619.45 | 43.75 | 1,575.70 |
| Signal-timing plan | 456.85 | 589.28 | 132.43 |
| The largest queue always | 1,156.05 | 300.90 | 855.15 |
| The largest queue first | 1,057.24 | 428.57 | 628.67 |
| At least $k$ vehicles each time | 1,294.14 | 127.17 | 1,166.97 |

Table 3: Traffic flow distribution $d\overline{X}$ concerning two road segments obtained by the application of state $E_2$ to the implemented algorithms within the range of $14,400$ steps in SUMO

One can observe that the *signal-timing plan* policy was the most widely distributed of all with $d\overline{X} = 132.43$. Secondly, we have *the largest queue first* policy with $d\overline{X} = 628.67$, then *the largest queue always* with $d\overline{X} = 855.15$, then *at least k vehicles each time* policy with $d\overline{X} = 1{,}166.97$, and finally, we have *the right of way* with 1,575.70.

Thus, we can conclude that the policy implemented in the *signal-timing plan* proved to be the best suited to control the passage of vehicles through the intersection between two paths with the traffic flow as defined by the state of traffic flow $E_2$.

In the following, we have the policy *the largest queue first*, this time, the second proved more suitable for the control of this type of traffic, following by *the largest queue always* which is in the third position. The next one is *at least k vehicles each time* following by *the way of right* policy, which is again, the least suitable for such transit.

### 4.2.3 Scenario Group $G_3$

Finally, Table 4 summarizes the results obtained by the application of state $E_3$ to the implemented algorithms within the range of $14,400$ steps in SUMO.

| Policy | $X_s$ | $X_w$ | $d\overline{X}$ |
|---|---|---|---|
| The right of way | 899.00 | 385.64 | 513.36 |
| Signal-timing plan | 751.75 | 233.57 | 518.18 |
| The largest queue always | 282.58 | 547.90 | 265.32 |
| The largest queue first | 227.80 | 545.51 | 317.71 |
| At least $k$ vehicles each time | 505.64 | 515.40 | 9.76 |

Table 4: Traffic flow distribution $d\overline{X}$ concerning two road segments obtained by the application of state $E_3$ to the implemented algorithms within the range of $14,400$ steps in SUMO

One can observe in this scenario that *at least k vehicles each time* policy is the most widely distributed of all with $d\overline{X} = 9.76$, reaching almost to equitable distribution between the two pathways, noting that the distribution of vehicles driving by the intersection of two roads in a given scenario tends to evenness as plus the difference of $d\overline{X}$ tends to zero. In second place, we have *the largest queue always* with $d\overline{X} = 265.32$, then *the largest queue first* with $d\overline{X} = 317.71$, then *signal-timing plan* with $d\overline{X} = 513.36$, and finally, we have *the right of way* with 518.18.

Thus, we can conclude that the policy implemented in the *at least k vehicles each time* clearly proved the most suitable to control the passage of vehicles through the intersection between two roads with traffic flow as defined by the state $E_3$.

In the following, we have *the largest queue always* that proved to be the second most suitable policy for this type of traffic, and *the largest queue first* is in the third one, *the way of right* in fourth position and, finally, we have *signal-timing plan* policy.

## 5. CONCLUSION AND FUTURE WORKS

This work demonstrated that the intersection control, typically implemented in Brazilian cities by traffic lights, can have a significant improvement with the application of algorithms based on the traffic flow. The proposition of mechanisms, in general, more suitable to intersection control is necessary, since efficient traffic management is a problem present and constant in our daily lives. As argued previously, the existing solutions are preferably based on signal timing plans with no adaptation. However, it is clear that different policies can be implemented to improve results.

Our target application involves transportation networks and urban mobility, issues that have aroused much interest in the whole contemporary society. And, in fact, solutions and mechanisms to improve urban mobility and transportation processes have been implemented and proposed, and are more affordable currently. Some examples are the adaptive traffic lights recently installed some Brazilian cities (e.g. Porto Alegre, Belo Horizonte e Fortaleza) and many ATIS (advanced traveler information systems) such as *Google Transit, Waze, Olho Vivo*, (from São Paulo, which provide to users information about public transport status), for instance.

More specifically, this paper presented a simulation in the context of transportation networks to deal with intersection control. Different policies for intersection management were evaluated. The simulation was configured with two roads: one from south to north ($SN$), and another from west to east ($WE$), and an intersection point $P$. Then three states of traffic flow were created: $E_{1-3}$, based on the preference of SN on WE at passing vehicles by $P$. Finally, we defined 15 different scenarios from the application of policies to defined traffic states. They were classified into three groups of scenarios according each traffic situation.

Experiments with these three scenarios were run to evaluate equitability. After experimentation, we may conclude that if the traffic is of the type defined by the state $E_1$, the policy implemented by the *the largest queue first* algorithm is the most suitable for control the passage of vehicles by $P$. While, when traffic is the type defined by the state $E_2$, the policy implemented by the *traffic lights* is the most suitable for the control of $P$. Finally, when the traffic of the type defined by the state $E_3$, implemented the policy by *at least k vehicles each time* is the most appropriate place for intersection control in $P$.

With these results, we can conclude that applying only one policy to intersection control is not the best solution. Since traffic is subject to dynamism and delays, different traffic scenarios need different policies. In this direction, this work represents a further step in efficient traffic signal control.

Future works include the implementation of more sophisticated policies. For instance, we could develop a hybrid policy, which is the junction of several policies. Furthermore, one can define other states that describe the traffic with more emphasis the realism in the simulation, taking into account other variables that influence the traffic flow, among others: variation of the maximum speed of the road, different types of vehicles and priorities (such as ambulances, firemen service), addition of passages through intersections, pedestrian accidents and other incidents that block and change the traffic flow.

Decentralized intersection control policies also need to be effectively proposed and evaluated before put them into practice. In this case, only V2V communication should be considered as well as the use of simulators target to transport networks and VANETs. Finally, such as in Vasirani & Ossowski [9], the transportation network can expand with the addition of new roads and lanes, and several intersections.

## Acknowledgments

This work was supported by the Brazilian Government agency FAPERGS/CNPq (Project RS-SOC - Rede Estadual de Simulação Social) under grant number 10/0049-7 and CTIC/RNP SIMTUR Cidades Inteligentes.

## 6. REFERENCES

[1] A. L. C. Bazzan. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multiagent Systems*, 18(3):342–375, June 2009.

[2] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. Sumo - simulation of urban mobility: an overview. In *Proceedings 3rd International Conference on Advances in System Simulation (SIMUL 2011). ThinkMind*, pages 63–68, October 2011.

[3] B. Chen and H. H. Cheng. A review of the applications of agent technology in traffic and transportation systems. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):485–497, June 2010.

[4] K. Dresner and P. Stone. Multiagent traffic management: a reservation-based intersection control mechanism. *Proceedings 3rd International Joint Conference o Autonomous Agents and Multiagent Systems*, pages 530–537, July 2004.

[5] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31:591–656, March 2008.

[6] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, and O. K. Tonguz. Self-organized traffic control. *Proceedings 7th ACM International Workshop on VehiculAr InterNETworking (VANET '10)*, pages 85–90, 2010.

[7] D. Krajzewicz, E. Brockfeld, J. Mikat, J. Ringel, C. Rossel, W. Tuchscheerer, P. Wagner, and R. Wosler. Simulation of modern traffic lights control systems using the open source traffic simulation sumo. In *Proceedings 3rd Industrial Simulation Conference 2005, EUROSIS-ET*, pages 299–302, 2005.

[8] M. Martin and P. Nurmi. A generic large-scale simulator for ubiquitous computing. In *Mobile and Ubiquitous Systems, Annual International Conference on*, pages 1–3, July 2006.

[9] M. Vasirani and S. Ossowski. A market-inspired approach to reservation-based urban road traffic management. *Proceedings 8th International Conference on Autonomous Agents and Multiagent Systems*, 1:617–624, July 2009.

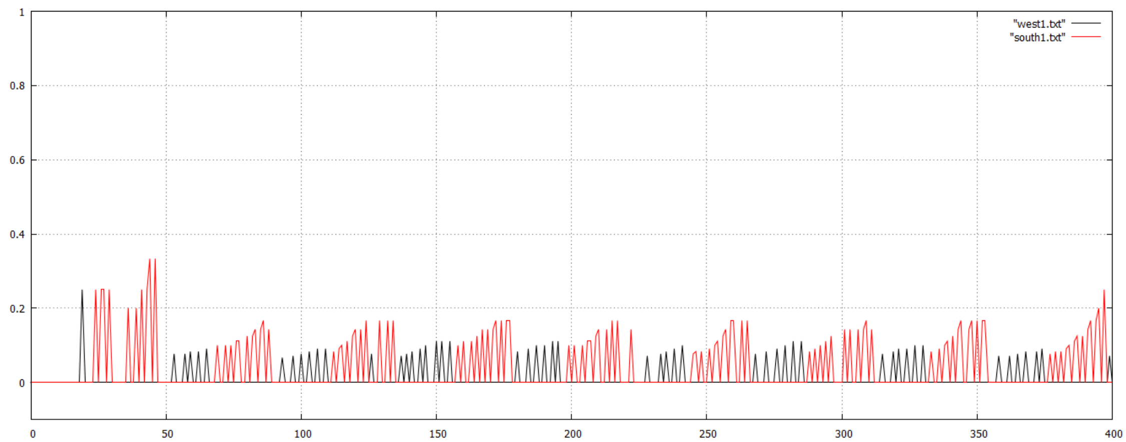



Figure 3: Scenario resulted from the application of *signal-timing plan policy* to the state of traffic flow $E_1$

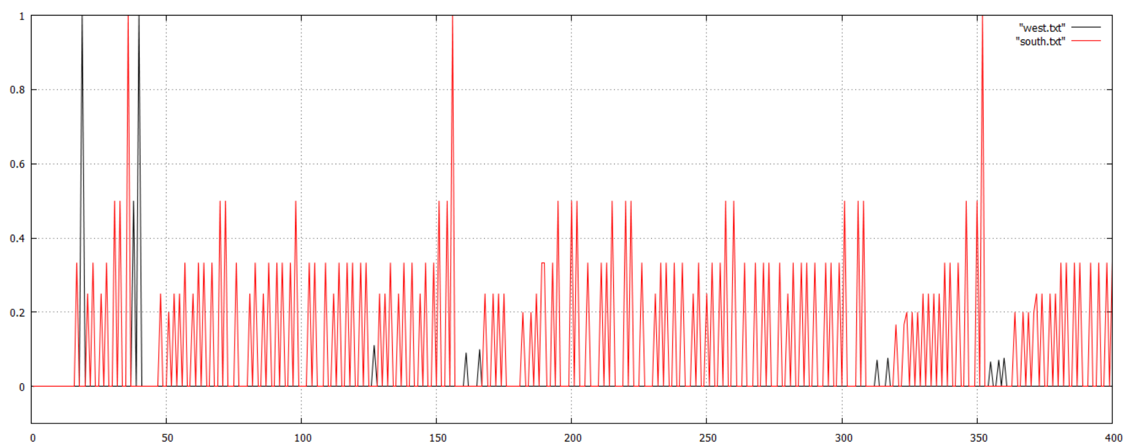



Figure 4: Scenario resulted from application of *the right of way policy* to the state of traffic flow $E_2$

# Overcoming Information Overload with Artificial Selective Agents: an Application to Travel Information Domain

Luis Macedo
University of Coimbra
Department of Informatics Engineering
3030 Coimbra, Portugal
macedo@dei.uc.pt

Hernani Costa
University of Coimbra
Department of Informatics Engineering
3030 Coimbra, Portugal
hpcosta@dei.uc.pt

Amilcar Cardoso
University of Coimbra
Department of Informatics Engineering
3030 Coimbra, Portugal
amilcar@dei.uc.pt

## ABSTRACT

We describe an application of Macedo's computational model of selective attention for overcoming the problem of information and interruption overload of intelligent agents in travel information systems. This computational model has been integrated into the architecture of a BDI artificial agent so that this can autonomously select relevant, interesting travel information of the (external or internal) environment while ignoring other less relevant information. The advantage is that the agent can communicate only that interesting, selective information to its processing resources (focus of the senses, decision-making, etc.) or to its human owner's processing resources so that these resources can be allocated more effectively. We illustrate and provide experimental results of this role of the artificial, selective attention mechanism in the travel domain.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

## General Terms

Algorithms

## Keywords

Information overload, Selective attention, Interest, Value of information, Surprise, Uncertainty, Resource-bounded agents, Personal agents

## 1. INTRODUCTION

The advent of information technology is a primary reason for the abundance of information with which humans are inundated, due to its ability to produce more information more quickly and to disseminate this information to a wider audience than ever before. Surprisingly, a lot of recent studies confirmed what Toffler [36] predicted a few decades ago: the overabundance of information instead of being beneficial is a huge problem having many negative implications not only in personal life but also in organizations, business, and in general in the world economy. In fact, research proves that the brain simply does not deal very well with a multitasking process [12]. This explains why decision quality and the rate of performing tasks degrades with increases in the amount of information being considered.

A fundamental strategy for dealing with this problem of information overload [24] should include making devices that incorporate themselves selective attention agents in order to decrease the amount of information considered in their own reasoning/decision-making processes or decrease the amount of information provided by them to humans, preventing these from a number of interruptions.

But how to model selective attention in artificial agents? Although selective attention has been thoroughly researched over the last 100 years in psychology and more recently in neuroscience (e.g., [10, 38]), at present there is no general theory of selective attention. Instead there are specific theories for specific tasks such as orienting, visual search, filtering, multiple action monitoring (dual task), and multiple object tracking.

In spite of this, a number of models of selective attention has been proposed in Cognitive Science (e.g., [9, 21]). Particularly related with these models is the issue of measuring the value of information. A considerable amount of literature has been published on these measures, especially from the fields of active learning and experimental design. Most of those measures rely on assessing the utility or the informativeness of information (e.g., [8, 20, 13, 33]). However, little attention has been given to the surprising and motive congruence value of information, giving the beliefs and desires of an agent.

Macedo, Reisenzein and Cardoso (e.g., [16, 19]), and Lorini and Castelfranchi [14] proposed, independently, computational models of surprise that are based on the mechanism that compares newly acquired beliefs to preexisting beliefs. Both models of artificial surprise were influenced by psychological theories of surprise (e.g., [23]), and both seek to capture essential aspects of human surprise (see for a comparison [18]).

In this paper we describe the application of Macedo's artificial selective attention mechanism [15] to travel information systems. In our approach, artificial agents of travel information systems make use of that mechanism so that only cognitively and affectively, interesting/relevant travel information is selected and forwarded to drivers. The selective attention mechanism relies on the psychological and neuroscience studies about selective attention which defend that variables such as unexpectedness, unpredictability, surprise, uncertainty, and motive congruence demand attention (e.g., [2, 10, 25]).

The next section presents an overview of Macedo's computational model of selective attention. Section 3 illustrates how this selective attention mechanism can be used for filtering irrelevant information in the travel domain. Section 4 examines the performance of the selective attention mechanism as well as its role on the decrease of unnecessary information. Finally, in Section 5 we present conclusions.

## 2. SELECTIVE ATTENTION AGENT

Selective attention may be defined as the cognitive process of selective allocation of processing resources (focus of the senses, etc.) on relevant, important or interesting information of the (external or internal) environment while ignoring other less relevant information. The issue is how to measure the value of information. What makes something interesting?

Macedo [15] developed previously an architecture for a personalized, artificial selective attention agent (see Figure 1). It is assumed that: (i) this agent interacts with the external world receiving from it information through the senses and outputs actions through its effectors; (ii) the world is described by a large amount of statistical experiments; (iii) the agent is a BDI agent [27], exhibiting a prediction model (model for generating expectations, i.e., beliefs about the environment), a desire strength prediction model (a model for generating desire strengths for all the outcomes of the statistical experiments of the world that are known given the desires of the agent – profile of the agent which include basic desires), as well as the intentions (these define the profile of the agent); (iv) the agent contains other resources for the purpose of reasoning and decision-making.

The first of the modules of the architecture (module 1 in Figure 1) is concerned with getting the input information. The second is the computation of the current world state. This is performed by generating expectations or assumptions (module 2), based on the knowledge stored in memory, for the gaps of the environment information provided by the sensors (module 1). We assume that each piece of information resulting from this process, before it is processed by other cognitive skills, goes through several sub-selective attention devices, each one evaluating information according to a certain dimension such as surprise (module 4), uncertainty (module 5), and motive-congruence/incongruence – happiness (module 6). For this task the selective attention mechanism takes into account some knowledge container (memory — preexisting information (module 7)), and the intentions and desires (motives — module 8). There is a decision-making module (module 9) that takes into account the values computed by those sub-selective attention modules and decides if a piece of information is relevant/interesting or not. Then, this module of decision-making selects the more relevant pieces of information so that other resources (reasoning, decision-making, displaying, communication resources, etc.) (module 10) can be allocated to deal with them.

The process of making the right decision depends heavily on a good model of the environment that surrounds agents. This is also true for deciding in which information should the agent focus. Unfortunately, the real world is not crystal clear to agents. Agents almost never have access to the whole environment, mainly because of the incompleteness and incorrectness of their perceptual and understanding components. In fact, it is too much work to obtain all the information from a complex and dynamic world, and it is quite likely that the accessible information suffers distortions. Nevertheless, since the success of agents depends heavily on the completeness of the information of the state of the world, they have to pursue alternatives to construct good models of the world even (and especially) when this is uncertain. According to psychologists, cognitive scientists, and ethologists [11, 26], humans and, in general, animals attempt to overcome this limitation through the generation of assumptions or expectations to fill in gaps in the present or future observational information. When the missing information, either of the present state of the world or of the future states of the world, becomes known to the agent, there may be an inconsistency or conflict between it and the assumptions or expectations that the agent has. As defended by Reisenzein [28], Gardenfors [7], Ortony and Partridge [25], etc., the result of this inconsistency gives rise to surprise which in our model of selective attention and according to previous studies plays a central role in selective attention. It also gives rise to the process of updating beliefs, called belief revision (e.g., [6]).

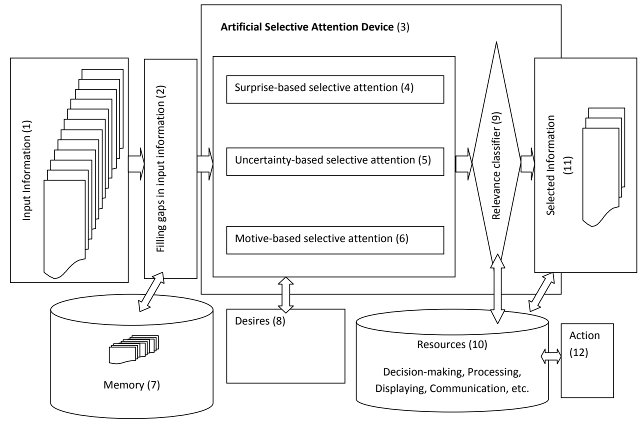



Figure 1: Architecture of an artificial selective attention agent.

The representation of the memory contents (beliefs) relies on semantic features or attributes much like in semantic networks [31] or schemas [30]. Each attribute, $attr_i$, viewed by us as a statistical experiment, is described by a probabilistic distribution, i.e., a set $A_i = \{< value_j, prob_j, desireStrength_j >: j = 1, 2, \ldots, n\}$, where $n$ is the number of possible values of the attribute, $P(attr_i = value_j) = prob_j$, and $desireStrength_j$ is the desirability of $attr_i = value_j$ (for a related work see [29]).

While the belief strengths are inferred from data using a frequentist approach and updated as new information is acquired, the desirability of the outcomes can be previously set up or learned based on the intentions and contexts of the agent on which it depends, suffering changes whenever the agent is committed with a new intention and/or in a new context. For modelling this dynamics, we make use a desire strength prediction model, i.e., a model for generating desire strengths for all the outcomes of the statistical experiments of the world that are know given the desires of the agent, the intentions, as well as the context of the user (for more details see [5, 4]). As seen before, the desire strength is associated with each attribute together with the belief strength.

Much like the motivation system of Clarion [35], the module of desires encompasses explicit (goals) and implicit motives (basic desires). Following the pluralist view of motivation [22, 32, 37], the sub-module of basic desires (basic motivations/motives) contains a set of basic desires that drive the behaviour of the agent by guiding the agent to reduce or to maximize a particular feeling [17]. Among the basic desires we can find surprise and curiosity.

The module of feelings receives information about a state of the environment and outputs the intensities of feelings. Following Clore [3], we include in this module affective, cognitive, and bodily feelings. The latter two categories are merged to form the category of non affective feelings. This means that this module is much broader than a module of emotion that could be considered. Feelings are of primary relevance to influence the behavior of an agent, because computing their intensity the agent measures the degree to which the desires are fulfilled. In this paper, we highlight the feelings of surprise and pleasantness/unplesantness.

Although the architecture of the computational model of selective attention includes all those above-mentioned sub-selective attention modules, we reserve some room in the architecture of the model for other sub-selective attention components, such as coping potential, complexity.

The next sub-sections describe each one of the dimensions for evaluating information, namely surprise, uncertainty, and motive congruence/incongruence. While the dimensions of surprise and uncertainty are related to the value of information to the belief store of the agent, the dimension of motive congruence/incongruence is related to the value of information to the goals/desires of the agent (these dimensions are related to the concepts of cognitive and affective feelings of [3] and belief-belief and belief-desire comparators of [29]).

## 2.1 Surprise Value of Information

We adopted the computational model of surprise of [16, 19] which is formally defined in Definition 1 (for related models see [18]). Macedo, Cardoso and Reisenzein computational model of surprise suggests that the intensity of surprise about an event $E_g$, from a set of mutually exclusive events $E_1, E_2, \ldots, E_m$, is a nonlinear function of the difference, or contrast, between its probability and the probability of the highest expected event $E_h$ in the set of mutually exclusive events $E_1, E_2, \ldots, E_m$.

Definition 1. *Let $(\Omega, A, P)$ be a probability space where $\Omega$ is the sample space (i.e., the set of possible outcomes of the experiment), $A = A_1, A_2, .., A_n$ is a $\sigma$-field of subsets of $\Omega$ (also called the event space, i.e., all the possible events), and $P$ is a probability measure which assigns a real number $P(F)$ to every member $F$ of the $\sigma$-field $A$. Let $E = \{E_1, E_2, \ldots, E_m\}$, $E_i \in A$, be a set of mutually exclusive events in that probability space with probabilities $P(E_i) >= 0$, such that $\sum_{i=1}^{m} P(E_i) = 1$. Let $E_h$ be the highest expected event from $E$. The intensity of surprise about an event $E_g$ from $E$ is given by:*

$$S(E_g) = \log(1 + P(E_h) - P(E_g)) \tag{1}$$

The probability difference between $P(E_h)$ and $P(E_g)$ can be interpreted as the amount by which the probability of $E_g$ would have to be increased for $E_g$ to become unsurprising.

## 2.2 Uncertainty-based Value of Information

Information is a decrease in uncertainty which, according to information theory, is measured by entropy [34]. When new information is acquired its amount may be measured by the difference between the prior uncertainty and the posterior uncertainty.

Definition 2. *Let $(\Omega, A, P_{prior})$ be a probability space where $\Omega$ is the sample space (i.e., the set of possible outcomes of the experiment), $A = A_1, A_2, .., A_m$ is a $\sigma$-field of subsets of $\Omega$ (also called the event space, i.e., all the possible events), and $P_{prior}$ is a probability measure which assigns a real number $P_{prior}(F)$ to every member $F$ of the $\sigma$-field $A$. Let $E = \{E_1, E_2, \ldots, E_m\}$, $E_i \in A$, be a set of mutually exclusive events in that probability space with probabilities $P_{prior}(E_i) >= 0$, such that $\sum_{i=1}^{m} P_{prior}(E_i) = 1$. Let $P_{post}$ be the posterior probability measure, after some data is acquired, which assigns a real number $P_{post}(F)$ to every member $F$ of the $\sigma$-field $A$ such that it assigns $P_{post}(E_i) >= 0$ with $\sum_{i=1}^{m} P_{post}(E_i) = 1$. According to information theory, the information gain of an agent after some data is acquired, $IG(E)$, is given by the decrease in uncertainty:*

$$\begin{aligned} IG(E) &= H_{prior}(E) - H_{post}(E) \\ &= -\sum_{i=1}^{m} P_{prior}(E_i) \times \log(P_{prior}(E_i)) - \\ &\quad (-\sum_{i=1}^{m} P_{post}(E_i) \times \log(P_{post}(E_i))) \end{aligned} \tag{2}$$

*$H_{post} = 0$ if and only if all the $P_{post}(E_i)$ but one are zero, this one having the value unity. Thus only when we are certain of the outcome does $H_{post}$ vanish, otherwise it is positive.*

*$IG$ is not normalized. In order to normalize it we must divide it by $\log(m)$ since it can be proved that $IG \leq \log(m)$:*

$$IG(E) = \frac{H_{prior}(E) - H_{post}(E)}{\log(m)} \tag{3}$$

## 2.3 Motive Congruence/Incongruence-based Value of Information

While the measure of surprise takes into account beliefs that can be confirmed or not, the pleasantness function that we describe in this subsection takes as input desires that, contrary to beliefs, can be satisfied or frustrated. Following the belief-desire theory of emotion [29], we assume that an agent feels happiness if it desires a state of affairs (a proposition) and firmly beliefs that that state of affairs obtains. The intensity of happiness about an event is a monotonically increasing function of the degree of desire of that event as formally defined in Definition 4.

Definition 3. *Let $(\Omega, A)$ be a measurable space where $\Omega$ is the sample space (i.e., the set of possible outcomes of the experiment) and $A = A_1, A_2, .., A_m$ a $\sigma$-field of subsets of $\Omega$ (also called the event space, i.e., all the possible events).*

*We define the measure of desirability of an event on* $(\Omega, A)$ *as* $D : A \rightarrow [-1, 1]$*, i.e., as a signed measure which assigns a real number* $-1 \leq D(F) \leq 1$ *to every member* $F$ *of the* $\sigma$*-field* $A$ *based on the profile of the agent, so that the following properties are satisfied:*

- $D(\emptyset) = 0$
- *if* $A_1, A_2, \ldots$ *is a collection of disjoint members of* $A$*, in that* $A_i \cap A_j = \emptyset$ *for all* $i \neq j$*, then*

$$D(\bigcup_{i=0}^{\infty} A_i) = \sum_{i=0}^{\infty} D(A_i) \tag{4}$$

*The triple* $(\Omega, A, D)$ *is called the desirability space.*

Definition 4. *Let* $(\Omega, A, P)$ *and* $(\Omega, A, D)$ *be the probability and the desirability spaces described, respectively, in Definition 1 and Definition 3. Let* $E = \{E_1, E_2, \ldots, E_m\}$*,* $E_i \in A$*, be a set of mutually exclusive events in that probability space with probabilities* $P(E_i) >= 0$*,* $\sum_{i=1}^{m} P(E_i) = 1$*. If* $P(E_g) = 1$*, the intensity of happiness, i.e., motive congruence, about an event* $E_g$ *from* $E$ *is given by:*

$$MC(E_g) = D(E_g) \tag{5}$$

## 2.4 The Principle of Selective Attention

Having defined the motive, the uncertainty-based, and surprise-based selective attention modules, we are now in a position to formulate, in a restricted sense (without the inclusion of other information measures such as complexity), the principle that a resource-bounded rational agent should follow in order to avoid an overabundance of information and interruptions in the absence of a model for decision-making. Note that if this model is known, the problem is reduced to the classical computation of the value of information that has been extensively studied (e.g., [8, 31]).

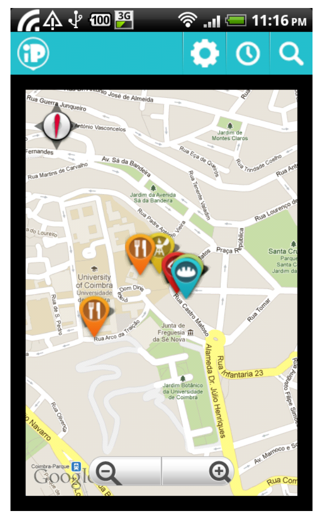
Definition 5. *A resource-bounded rational agent should focus its attention only on the relevant and interesting information, i.e., on information that is congruent or incongruent to its motives/desires, and that is cognitively relevant because it is surprising or because it decreases uncertainty.*

We may define real numbers $\alpha$, $\beta$, and $\gamma$ as levels above which the absolute values of motive congruency, surprise, and information gain (decrease of uncertainty), respectively, should be so that the information can be considered valuable or interesting. These are what we called the triggering levels of alert of the selective attention mechanism. Note that, making one of those parameters null is equivalent to removing the contribution of the corresponding component from the selective attention mechanism (for a different approach see Martinho and Paiva's attention grabbing mechanism [21] whose main feature is not relying on tuned parameters but on expectation and prediction error).

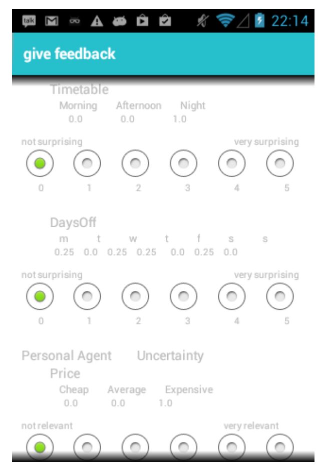
# 3. PRACTICAL APPLICATION

The Selective Attention-based, Multi-Agent, Travel Information System architecture (see Figure 4) we developed involves a master agent and personal agents (for related works on this domain see [1]). There is a personal selective attention agent for each registered traveler. Each personal agent models an user cognitively and motivationally and acts on his/her behalf, i.e., each personal agent has information about the expectations and desires of its owner based on their travel history. The main role of the master agent is collecting information from several information sources and sending it to the personal agents so that they can selectively deliver information to the several mobile devices owned by humans.

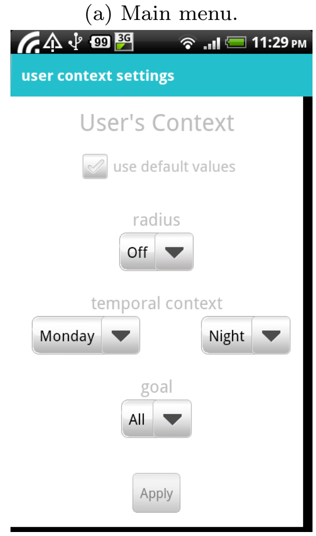
Physically, the master and the personal agents might inhabit in the same machine. This is the case of our system: there is a server that accommodates both the master agent and the personal agents. There is also an interface of the personal agents that acts as a client and which is stored in mobile devices owned by humans (see Figure 2).



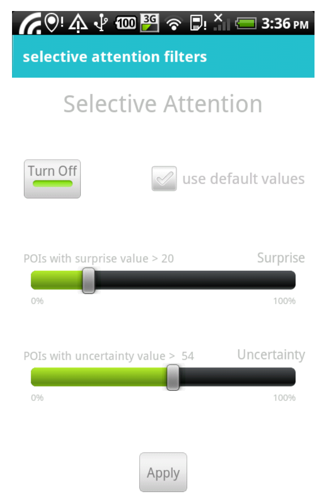

(a) Main menu.




(b) Feedback menu.




(c) User context settings.




(d) Selective attention filters.

Figure 2: iPOIs interface.

The `Master Agent` is responsible for starting, not only the Web Agents, but also the Personal Assistant Agents (PAAs), described in Figure 4 as $\texttt{PAA}_1 \cdots \texttt{PAA}_n$. The `Master Agent` is also responsible to reply the `PAAs` when they ask for information about a specific POI. Although the system is capable of retrieving POIs' information from several location-based services such as Foursquare API[1] (a location-based social network) and Bing Traffic API[2] (that provides information about traffic incidents and issues, e.g., construction sites and traffic congestion), for the purpose of this work only the Foursquare service is used, which explains why in this work we used only one Web Agent ($\texttt{WA}_{\texttt{foursquare}}$). As it can be seen in Figure 2a, the system shows all the POIs retrieved from the system, taking into account the current user's context and intention (Figure 3), as well as his/her selective attention preferences (Figure 2d). Clicking in each POI's icon, the user can see an information window with the expected surprise and information gain values associated to the price, schedule and day(s) off. One of the most relevant feature of this interface is the menu presented in Figure 2b, where the user is allowed to give feedback about the expectations of his PAA.

$\texttt{WA}_{\texttt{foursquare}}$ implements several methods available through the Foursquare API[3], allowing it to start requesting for POIs in a pre-defined geographical area. During this process, it filters out all the POIs that do not belong to the categories of concern, and stores the remaining POIs in the system's database (presented in Figure 4 as `POIs Database`). This autonomous agent is constantly searching for new information, and verifying if the data stored in the database is up-to-date.

Context is the key to personalise recommendations made by the PAAs for their users. Thus, a set of attributes need to be defined in order to characterise the `POI`'s context, as well as the `user`'s context and intentions. Since these attributes need to be combined, an Android application, named iPOIs, was been created to this purpose, i.e., to show the current user location, his context and intention. The main attributes used to define the user, the POI and the information available in the interface are shown in Figure 3.

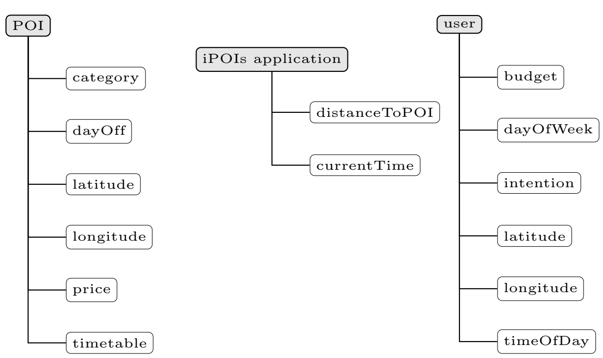



Figure 3: Main attributes used to define the context of the user, POI and the iPOIs application.

Possible values for each attribute of the `POI`'s context are:

- `category = {food, shopping, nightlife}`, actually we use the `sub-category, e.g., food =` ⟨`sandwichShop, vegetarian, etc.`⟩, `shopping =` ⟨`men'sApparel, women'sApparel, etc.`⟩ `and nightlife =` ⟨`wineBar, disco, etc.`⟩
- `dayOff = {a day of the week or combinations}`

[1]https://developer.foursquare.com/
[2]http://msdn.microsoft.com/en-us/library/hh441725
[3]`https://developer.foursquare.com`

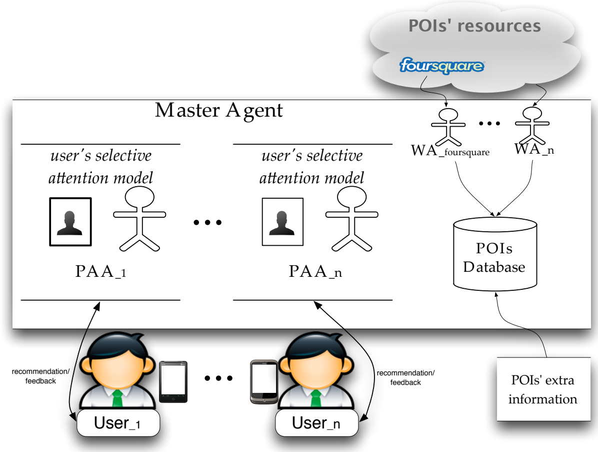



Figure 4: System's Architecture.

- `price = {cheap, average, expensive}, e.g., for lunch {cheap≤5€; 5€>average≤7€; expensive>7€}`
- `timetable = {morning, afternoon, night, or combinations}`

Possible values provided by the `iPOIs interface` are:

- `distanceToPOI = {near≤` $200m$`;` $200m$ `>average≤` $300m$`; far>` $300m$`}`
- `currentTime = {current day of the week and period of the day (morning, afternoon or night)}`

Possible values for each attribute of the `user`'s context are:

- `budget = {low, medium, high}, e.g., for lunch {low≤5€; 5€>medium≤7€; high>7€}`
- `dayOfWeek = {current day of the week}`
- `intention = {coffee, lunch, dinner, party}, e.g., drink coffee in a {bakery, coffeeShop, etc.}, have lunch and dinner in {burgers, BBQ, etc.} and party in a {bar, disco, etc.}`
- `timeOfDay = {morning, afternoon or night}`

Let us illustrate how the value of information is computed by the selective attention mechanism. Suppose that a traveller's navigation system provided the information of a specific POI, a restaurant denoted by $A$, for an agent (that represents a driver) based on its profile (e.g., preference for cheap restaurants). Suppose the agent has the following expectations for the price of POI $A$, for a certain period/time of the day for a certain day of the week: 60% of probability of "low price" (event $E_1$), 30% of probability of "moderate price" (event $E_2$), and 10% of probability of "high price" (event $E_3$). Suppose the desire strengths of these events are 1, -0.5, and -1, respectively. What is the relevance of becoming aware that the price of restaurant A is low (event $E_1$)? Considering solely the motive-based component, the outcomes (events $E_1$, $E_2$, and $E_3$) elicits happiness (motive congruence) with intensity 1, -0.5 and -1, respectively. $E_1$ is congruent/consistent with the goals of the agent, while $E_2$ and $E_3$ are incongruent with the goals of the agent.

According to Equation 1, the surprise value of $E_1$, $E_2$, and $E_3$ are, respectively, 0, 0.38, and 0.58. Illustrating for the case of $E_3$:

$$Surprise(E_3) = \log(1 + P(E_1) - P(E_3)) = \log(1 + 0.6 - 0.1) = 0.58 \quad (6)$$

According to Equation 3, the normalized information gain value of $E_1$, $E_2$, or $E_3$ is:

$$IG(E) = \frac{H_{prior}(E) - H_{post}(E)}{\log(m)} = \frac{H_{prior}(E) - 0}{\log(3)} = \frac{-\sum_{i=1}^{3} P_{prior}(E_i) \times \log(P_{prior}(E_i))}{\log(3)} = 0.82 \quad (7)$$

Assume the Principle of Selective Attention described above, with parameters $\alpha = 0.3$, $\beta = 0.5$, and $\gamma = 0.6$. Are all these events interesting? Considering the motive-based component all those events are interesting. However, from the perspective of the surprise-based selective attention component, the answer is "no" to the question related with the events $E_1$ and $E_2$ in that their surprise values, 0 and 0.38, respectively, are below $\beta$. With respect to $E_3$ the answer is "yes" given that its surprise value is 0.58. Taking the uncertainty-based component into account, the answer is "yes" for all the events because their occurrence gives a normalized information gain of 0.82 which is above $\gamma$.

By filtering out information that seems to be uninteresting, the selective attention mechanism prevents an agent (and also its owner – a driver in this case) from being interrupted so many times as in the absence of the selective attention mechanism and consequently prevents its reasoning/decision-making resources from dealing with irrelevant information. But, is the quality of the decisions of the driver affected by not receiving that presumably irrelevant information? In other words, was the suppressed information erroneously considered as irrelevant? If the answer is "yes", we have a false negative. This error occurs when we are making a negative inference which is actually true. In the above example, if the information of the occurrence of $E_3$ was not revealed, the driver would have stopped and enter restaurant $A$ that might be less useful than an alternative. The reverse can also happen: was the provided information erroneously considered as relevant? If the answer is "yes", we have a false positive or false alarm. This error occurs when we are making a positive inference which is actually false. This problem of knowing the correctness of preventing an interruption is quite similar to errors type I and II of statistical hypothesis testing. A reasonable empirical way to answer these questions is by comparing the classifications of the selective attention agent to those of humans. This is the main goal of the experiment described in the next section.

## 4. EXPERIMENT

We conducted an experiment to evaluate the performance and the potential benefits of the personal selective attention agent for filtering unnecessary information for its owner (a human traveler). To do that we assessed its performance considering the opinions of the human travelers, comparing their classifications about whether some information is relevant or not and the classifications of the selective attention agent. The selective attention agent is considered to perform erroneously if it filters a relevant information or if it does not filter an irrelevant information.

The experimentation was performed in downtown of the city of Coimbra, Portugal, which is characterized by a high density and diversity of POIs. Furthermore, the type of POIs considered were restricted to `{Food, Shopping, Nightlife}` which are among the more frequent categories in that region of the city. The number of sub-categories for `Food` are `44`, `Shopping 8` and `Nightlife 11`, with `271`, `10` and `84` different POIs, respectively. The extra information manually gathered from these `365` places was the POI's price, the day off and the timetable.

This experiment can be divided into three different evaluations. Firstly we made a manual evaluation, to analyse the true relevance of the recommended POIs, and calculated the exact agreement between the human judges. Then, we performed a correlation analysis to compare the selective attention values given by the PAAs with those of the manual evaluation. Finally, we analysed the system performance.

To test our approach, we used a set of real scenarios. More precisely, in this experiment we used three different locations with higher POIs density. The information of these different locations was combined with different situations (i.e., different user's contexts and intentions). Each one of these combinations is called a *run*. For instance, $r_1$ = `[40.208934, -8.429067, Morning, Sunday, Coffee]` represents one of those runs in which it can be seen the user's GPS location, time of day, day of the week and intention/goal. In this experiment, we analysed 13 runs in a total of 65 evaluated POIs[4]:

- 5 runs, goal: drink a coffee (25 evaluated POIs);
- 2 runs, goal: have lunch (10 evaluated POIs);
- 3 runs, goal: have dinner (15 evaluated POIs);
- 3 runs, goal: go party (15 evaluated POIs).

To perform this evaluation, we used the interface of the iPOIs application, illustrated in Figure 2.

We asked 9 human judges to rate some POIs attributes about the surprise and uncertainty-based value. The attributes evaluated in the experiment were the POI's price, the POI's schedule, the POI's day(s) off and the POI as a whole. Each human judge was asked to assign one value to these attributes, considering their information gain and surprise, using the scale ∅ to 5, where ∅ means that there was no information gain (regarding its uncertainty-based value) or no surprise (regarding the surprise intensity). We then calculated the exact agreement (`EA`) between the human judges used to calculate the coefficient correlation with the values of surprise and information gain computed by the artificial agents.

The `EA` among the judges (`EA`: $0\% \leq$ `EA` $\leq 100\%$), for all the data evaluated, is presented in Table 1, where the attributes price, schedule, day(s) off and all the attributes together are presented as `Price`, `Sche.`, `D.Off` and `All`, respectively.

Table 1: Exact agreement between the human judges.

| **Information gain** | | | | **Surprise** | | | |
|---|---|---|---|---|---|---|---|
| **Price** | **Sche.** | **D.Off** | **All** | **Price** | **Sche.** | **D.Off** | **All** |
| 100 | 98.58 | 97.61 | 100 | 100 | 97.61 | 96.64 | 100 |

The parallelism between the exact agreement of humans

[4]This evaluation, in average, took approximately 1 hour.

and the uncertainty and surprise-based values computed by the artificial agent about the POIs' attributes was quantified by Spearman's coefficient. This correlation coefficient give us an idea on how these variables are correlated.

Table 2 shows these correlation coefficients between the `EA`, given by the human judges, and the uncertainty and surprise-based values computed by the artificial agent for the four types of attributes considered, through the 13 runs. As it can be seen, the results are promising. Although this means that there exists a positive correlation in general, some of them do not have a strong correlation value (for instance, the surprise value for the POI's attribute price). This happens due the fact that the price was not so surprising to the judges than the day(s) off. For example, when the agent presents similar surprise expectation values to cheap and average and the POI's price is cheap, the judges do not gave a high surprising value to this information. On the other hand, the judges gave a high surprise value when the POI is closed and the agent presents a low surprise value to that specific day(s) off. The opposite occurs to the importance that the judges gave to the uncertainty-based value of the price.

Table 2: Correlation between the `EA` and the selective attention models.

| Information gain | | | | Surprise | | | |
|---|---|---|---|---|---|---|---|
| **Price** | **Sche.** | **D.Off** | **All** | **Price** | **Sche.** | **D.Off** | **All** |
| 0.8459 | 0.4036 | 0.4218 | 0.6321 | 0.2557 | 0.5811 | 0.5218 | 0.4901 |

Finally, in the third part of the experiment, we performed an information retrieval task, where the uncertainty and the surprise components (named $\alpha$ and $\beta$, respectively) was used to analyse the system's performance. To do that, we used the $\alpha$'s and $\beta$'s average (i.e., $\overline{\alpha}$ and $\overline{\beta}$), from the 13 runs, for the four POI's attributes analysed in this work. To measure the quality and the quantity of POIs correctly selected, precision, recall and $F_1$ were computed in the following manner:

$$Precision = \frac{Selected\_correct\_POIs}{Selected\_POIs} \tag{8}$$

$$Recall = \frac{Selected\_correct\_POIs}{Total\_correct\_POIs} \tag{9}$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

For each component, Table 3 presents the resulting precision, recall and $F_1$ scores (expression 8, 9 and 10) and the respective $\alpha$'s and $\beta$'s used, where the attributes price, schedule, day(s) off and all the attributes together are presented as `Price`, `Sche.`, `D.Off` and `All`, respectively.

As it can be seen, the selective attention components performed similarly regarding the distinct attributes. Nevertheless, the $F_1$, on average, for the $\alpha$ component is higher than the $F_1$ average for the $\beta$ (≈74.90% and ≈62.36%, respectively), which means that the uncertainty-based component performs better than the surprise component on average. Even though some of the $F_1$ values show low performance (e.g., the attribute price with $\beta$=0.9542 (34.78%)), most of them achieve high $F_1$ (e.g., the attribute price with the $\alpha$=0.0625 (93.75%) or the attribute schedule with the $\beta$=0.9180 (81.97%)). These results are promising, supporting the idea of applying a computation model of selective attention into location-based services, as an alternative or an extension of traditional recommender systems.

Table 3: System's performance for the two selective attention components, with their respective $\alpha$ and $\beta$.

| | | **Precision (%)** | **Recall (%)** | $F_1$ **(%)** |
|---|---|---|---|---|
| *Price* | $\alpha$=0.0625 | 90.90 | 96.68 | 93.75 |
| | $\beta$=0.9542 | 24.24 | 61.54 | 34.78 |
| *Sche.* | $\alpha$=0.0975 | 93.93 | 52.54 | 67.39 |
| | $\beta$=0.9180 | 75.75 | 89.29 | 81.97 |
| *D.Off* | $\alpha$=0.0469 | 96.97 | 55.17 | 70.33 |
| | $\beta$=0.9342 | 60.61 | 83.33 | 70.18 |
| *All* | $\alpha$=0.0646 | 93.94 | 53.45 | 68.13 |
| | $\beta$=0.9429 | 45.45 | 100 | 62.50 |

# 5. CONCLUSIONS

We presented an approach for filtering unnecessary information. We found evidence indicating that the mechanism contributes for decreasing the amount of unnecessary information while maintaining acceptable the performance of the owner (a human).

Besides, agents equipped with a selective attention filter can be successful personal assistants of humans, integrated for instance in mobile devices, so that their human users are prevented from unnecessary interruptions. This may be of high value in critical situations such as driving a car in that, as reported by [8], numerous cognitive studies have provided evidence of the problems in information processing exhibited by humans when dealing with large amounts of information such as that the speed at which humans perform tasks drops as the quantity of information being considered increases, and that the rate of performing tasks can be increased by filtering irrelevant information.

# 6. REFERENCES

[1] A. Bazzan and F. Kluegl. A review on agent-based technology for traffic and transportation. *Knowledge Eng. Review*, 2013.

[2] D. Berlyne. *Conflict, arousal and curiosity.* McGraw-Hill, New York, 1960.

[3] G. Clore. Cognitive phenomenology: Feelings and the construction of judgment. In L. Martin and A. Tesser, editors, *The Construction of Social Judgments*, pages 133–163. Lawrence Erlbaum Associates, Hillsdale, NJ, 1992.

[4] H. Costa, B. Furtado, D. Pires, L. Macedo, and A. Cardoso. Context and Intention-Awareness in POIs Recommender Systems. In *6th ACM Conf. on Recommender Systems, 4th Workshop on Context-Aware Recommender Systems*, RecSys'12, page 5, Dublin, Ireland, September 2012. ACM.

[5] H. Costa, B. Furtado, D. Pires, L. Macedo, and A. Cardoso. Recommending POIs based on the User's Context and Intentions. In *Highlights on Practical Applications of Agents and Multi-Agent Systems*, volume 365 of *Communications in Computer and*

*Information Science*, pages 166–177. Springer, Salamanca, Spain, May 2013.

[6] P. Gärdenfors. Belief revision: An introduction. In P. Gärdenfors, editor, *Belief Revision*, pages 1–20. Cambridge University Press, Cambridge, UK, 1992.

[7] P. Gärdenfors. The role of expectations in reasoning. In M. Masuch and L. Polos, editors, *Knowledge Representation and Reasoning Under Uncertainty*, pages 1–16. Springer-Verlag, Berlin, 1994.

[8] E. Horvitz and M. Barry. Display of information for time-critical decision-making. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 296–305. Morgan Kaufmann, 1995.

[9] E. Horvitz, A. Jacobs, and D. Hovel. Attention-sensitive alerting. In *Proceedings of the Fifteenth Conference on Uncertainty and Artificial Intelligence*, pages 305–313. Morgan Kaufmann, 1999.

[10] D. Kahneman. *Attention and effort*. Prentice-Hall, Englewood Cliffs, NJ, 1973.

[11] C. Kline. *Observation-based expectation generation and response for behavior-based artificial creatures*. Msc thesis, Massachusetts Institute of Technology, 1999.

[12] T. Klingberg. *The overflowing brain: information overload and the limits of working memory*. Oxford University Press, 2008.

[13] D. Lindley. On a measure of information provided by an experiment. *The Annals of Mathematical Statistics*, pages 986–1005, 1955.

[14] E. Lorini and C. Castelfranchi. The cognitive structure of surprise: looking for basic principles. *Topoi: An International Review of Philosophy*, 26(1):133–149, 2007.

[15] L. Macedo. A computational model for forms of selective attention based on cognitive and affective feelings. In U. Drewitz, N. Russwinkel, H. R. Russwinkel, and H. Rijn, editors, *Proceedings of the International Conference on Cognitive Modelling (ICCM 2012)*, pages 145–150. Technische Universität Berlin., 2012.

[16] L. Macedo and A. Cardoso. Modelling forms of surprise in an artificial agent. In J. Moore and K. Stenning, editors, *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 588–593. Erlbaum, Edinburgh, Scotland, UK, 2001.

[17] L. Macedo and A. Cardoso. Exploration of unknown environments with motivational agents. In N. Jennings and M. Tambe, editors, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 328 – 335. IEEE Computer Society, New York, 2004.

[18] L. Macedo, A. Cardoso, R. Reisenzein, E. Lorini, and C. Castelfranchi. Artificial surprise. In J. Vallverdú and D. Casacuberta, editors, *Handbook of Research on Synthetic Emotions and Sociable Robotics: New Applications in Affective Computing and Artificial Intelligence*, pages 267–291. IGI Global, Hershey: USA, 2009.

[19] L. Macedo, R. Reisenzein, and A. Cardoso. Modeling forms of surprise in artificial agents: empirical and theoretical study of surprise functions. In K. Forbus, D. Gentner, and T. Regier, editors, *Proceedings of the 26th Annual Conference of the Cognitive Science Society*, pages 873–878. Lawrence Erlbaum Associates, Inc., Chicago, Illinois, USA, 2004.

[20] D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[21] C. Martinho. Using anticipation to create believable behaviour. In *In: Proceedings of the AAAI*, pages 175–180. Press, 2006.

[22] W. McDougall. *An introduction to social psychology*. Methuen, London, 1908.

[23] W. U. Meyer, R. Reisenzein, and A. Schützwohl. Towards a process analysis of emotions: The case of surprise. *Motivation and Emotion*, 21:251–274, 1997.

[24] N. O'Connell. Interruption overload. *Strategic Direction*, 24(10):3–5, 2008.

[25] A. Ortony and D. Partridge. Surprisingness and expectation failure: what's the difference? In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 106–8. Morgan Kaufmann, Milan, Italy, 1987.

[26] J. Piaget. *The origins of intelligence in children*. International Universities Press, New York, 1952.

[27] A. Rao and M. Georgeff. BDI agents: from theory to practice. In *Proceedings of the First International Conference on Multiagent Systems*, pages 312–319. MIT Press, San Francisco, CA, USA, 1995.

[28] R. Reisenzein. The subjective experience of surprise. In H. Bless and J. Forgas, editors, *The message within: The role of subjective experience in social cognition and behavior*. Psychology Press, Philadelphia, PA, 2000.

[29] R. Reisenzein. Emotions as metarepresentational states of mind: Naturalizing the belief-desire theory of emotion. *Cognitive Systems Research*, 9, 2008.

[30] D. Rumelhardt and A. Ortony. The representation of knowledge in memory. In R. Anderson, R. Spiro, and W. Montague, editors, *Schooling and the Acquisition of Knowledge*, pages 99–135. Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

[31] S. Russell and P. Norvig. *Artificial intelligence - a modern approach*. Prentice Hall, Englewood Cliffs, NJ, 3rd edition, 2010.

[32] S. H. Schwartz. Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries. In M. P. Zanna, editor, *Advances in experimental social psychology*, volume 25, pages 1–65. Academic Press, NY, 1992.

[33] B. Settles. *Curious Machines: Active Learning with Structured Instances*. PhD thesis, University of Wisconsin, Madison, USA, 2008.

[34] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, 1948.

[35] R. Sun. The motivational and metacognitive control in clarion. In W. Gray, editor, *Modeling Integrated Cognitive Systems*. Oxford University Press, 2007.

[36] A. Toffler. *Future Shock*. Bantam Book, 1970.

[37] B. Weiner. *Human motivation*. Holt, Rinehart & Winston, New York, 1980.

[38] R. D. Wright and L. M. Ward. *Orienting of Attention*. Oxford University Press, Oxford, UK, 2008.

# Towards an Agent Coordination Framework for Smart Mobility Services

Andrea Sassi and Franco Zambonelli
Dipartimento di Scienze e Metodi dell'Ingegneria
Università di Modena e Reggio Emilia, Italy
name.surname@unimore.it

## ABSTRACT

Smart and social mobility services will soon hit the streets of our cities. However, most of existing solutions so far are built through different operations that don't lie on the same processing flow, neither don't share with each others their input data streams. The understanding of how to design a general-purpose framework, supporting a variety of integrated services and promoting direct users involvement, is still missing. In this paper, we first show our conceptual vision of smart mobility services, focusing on the cooperation and interoperability of the actors involved. We then analyze the infrastructural requirements to enable such smart mobility services and present the characteristics of a general-purpose framework for the provisioning of smart mobility services, conceived as a distributed and open agent coordination infrastructure. To exemplify, we show how the framework can be applied in the context of an urban ride-sharing service.

## Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Coherence and coordination, Multiagent systems

## General Terms

Algorithms, Design

## Keywords

Socio-technical System, Pervasive Computing, Smart Mobility Services, Agents Coordination, Ride-sharing

## 1. INTRODUCTION

The dramatic progress in embedded and mobile computing technologies, smart phones in primes, along with the pervasive diffusion of social networking tools, let us envision the emergence of a dense networked ICT infrastructure. In such infrastructure, coordinated human agents (i.e., the citizens) and software/hardware agents will interact with each other in such infrastructure so as to serve – at the same time – individual-level and urban-level goals, as if they were part of a single socio-technical system.

The overall behavior of such system will be driven by a variety of urban services which aim to improve the overall quality of life of individuals by providing them with tools to better interact with the urban environment, and also by shaping the activities of the urban environment itself, to suit their own needs.

One can consider a completely distributed software architecture deployed over individuals on their smart phones and over hardware sensors and actuators. However, a centralized entity able to continuously monitoring and redirecting the behavior of the agents will facilitate the dealing with city-scale problems.

The future pervasive urban services will be supported by bringing at work together the complementary sensing, computing, and actuating capabilities of the interconnected agents, and by closing them in a feedback loop (see Figure 1). After an initial learning phase in which raw data from sensors are collected, processed and classified, the agents will be skilled with context inference and anticipatory computing capabilities, as examples, and they will suggest tailored recommendations to the hardware actuators and to themselves. Closing these capabilities in a loop lets measure the goodness and the adoption rate of the suggested recommendations, by making clear their causal relation with the effects they generate. The process results in the generation of awareness, which can describe both individual and collective characters, related respectively to single agents and to a collection of those [3].

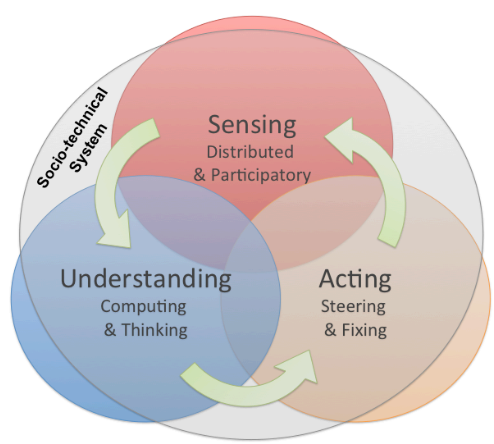



**Figure 1: The sensing-understanding-acting feedback loop enabled by agents coordination.**

This work focuses on smart mobility services enabled through the sensing-understanding-acting activities of the agents in the improvement of urban mobility. That is, to increase the effectiveness of individual mobility while at the same time improving the overall urban mobility (see Figure 2).

Human agents will play a fundamental role in the deployment of mobility services, since they can act both as consumers and as providers (e.g., via their private cars or simply by supplying information) of the services. Human social interactions can be pushed through a precise dynamic orchestration of the enabled data streams coming from both humans and ICT devices.

Our contributions are grounded on presenting how, in the scenario introduced above, the provisioning of integrated smart mobility services, can be effectively realized by a specifically suited coordination framework. Such coordination framework will be proposed as capable of supporting the iterative closed process of:

- Detecting mobility events related to the moving agents on the infrastructure, by harnessing the surrounding portion of the mobility data network shaped by the infrastructure itself, and also by processing the stream of incoming requests for mobility services;
- Identifying the possible solutions to satisfy expressed mobility needs based on the current state of things and of requests; anticipate future situations and future (or latent) mobility needs;
- Putting in act the necessary actions on actuator agents, or persuade human to act in certain ways, so as to end up realizing a coherent and sustainable set of services to satisfy the recognized needs.

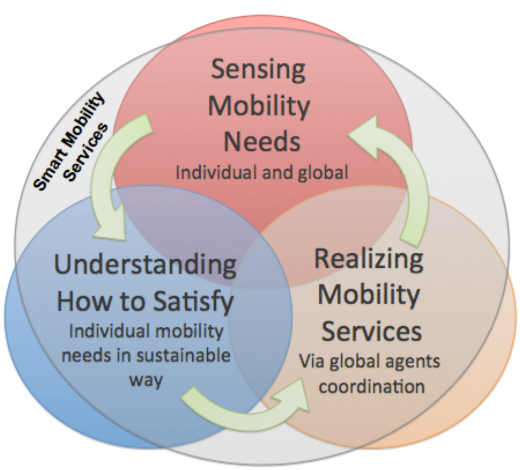



**Figure 2: Smart mobility services enabled by agents coordination.**

We believe, a general-purpose coordination framework that supports the shaping and the provisioning of smart mobility services will represent a powerful tool for urban designers and city administrators to make urban mobility services more efficient in terms of cost of the infrastructure (by harnessing the same sensors and actuators that self-reconfigure themselves upon specific requests, as well the same software architecture) and amount of data collected and processed (by sharing them among services with different purposes, instead of replicating similar operations for each service). Furthermore the provisioning of integrated solutions for different mobility needs can increase their individual adoption rate, towards the aim of reaching a critical mass of users, and thus can increase their effectiveness.

The contributions of this paper are to introduce our conceptual vision of what smart mobility services can be (Section 2), identify a set of infrastructural requirements for a general-purpose agent coordination framework (Section 3), sketch a conceptual model of the coordination framework for smart mobility services (Section 4), and introduce a use case in the area of ride-sharing (Section 5).

The paper also shortly discusses related works (Section 6) before concluding (Section 7).

## 2. SMART MOBILITY SERVICES

### 2.1 From ITS to smart mobility services

The recent dramatic progresses in ICT technologies, have led to the emergence of a very broad area of research in *Intelligent Transportation Systems* (ITS). ITS, in general, represent the most advanced way to establish a real-time transportation management, and consists in harnessing ICT technologies to better address users mobility needs and to support urban authorities decisions [1, 29].

ITS aim to improve urban transport performance, and can address in turns the problems and issues of pedestrians, cyclists, private vehicles, public transports, and roadside infrastructures. However, the application of ITS is often limited to the provisioning of on-demand web-services, with little or no interactions between users and contributions from user themselves. Furthermore, ITS do not offer a unified and integrated approach to support urban mobility in all its aspect, and often they own independent approaches for different mobility needs.

In general, the shift from ITS to *smart mobility services* must pursue the desired comfort for citizens and the satisfaction for urban authorities at the same level, by improving traffic efficiency and road capacity on the transportation network at an integrated, global, level. The services focus to impact on the development of increased social participation of citizens, where they are no longer simply requestors of mobility services, but can in turn play a role in the provisioning of services. Such an endeavor can feed cooperation and sharing practices with incentives and regulations.

Smart mobility services consist of all the mobility solutions enabled by pulling data from the available set of agents, generating higher information out of them, and enabling potential social interactions between a set of agents. The utility information is returned to them in such a way as to reinforce their interaction.

Citizens with mobility needs receive recommendations built on the matches with the services provided by other citizens, thanks to the supporting ICT infrastructure. Such recommendations can be strengthen if users have a similar profile, especially in terms of collaborative behavior. Data from social networks can detect social communities with same interests and mobility habits [5, 33]. The system will monitor the eventual adoption of the recommendation, and its effectiveness (was the service actually available?). Finally, it will update the profiles of the involved agents, to provide more useful recommendations in the future.

## 2.2 Example of smart mobility services

Let us now see some examples of such smart mobility services:

- **Parking Match.** A driver is approaching her destination and tries to find a vacant parking space. Some time earlier, another driver has left a parking lot in the same area. A parking match takes place and the driver is reached by a parking recommendation. Data involved in the matching process can come directly from the users involved, from the parking sensors installed on the infrastructure, or on users vehicles [22, 21].

- **Itinerary Match.** Consider the concurrent presence of the same users in a given set of locations at different times. When a spatio-temporal analysis on the data reveals that such co-location happens regularly (as seen in [8, 16]), it identifies a possible pool of commuters that make similar trips. The system should persuade them to switch to carpooling, making them aware of the benefits they have. Available carpooling services show how struggling is to reach a critical mass, hence social incentives are crucial (some carpooling issues are presented here [14]).

- **Taxi Match.** A taxi is hailed on the street by a person. While the driver is moving towards client's destination, he shares his route with other people that are looking for a ride (as described here [20]). If someone with a compatible trip ask for a ride, then taxi service becomes shared. Thus, its cost is lower for the clients and the revenue increase for the taxi driver. This service could seem similar to the previous one, but it mainly differs in terms of how the matches take place. The Itinerary Match mainly evaluates historical trips and habits, the latter considers real-time data.

- **Multimodal Rides Match.** A person explicitly declares a destination from her starting location, asking for directions. A selection of a spatio-temporal portion of data streams occurs. Multimodal directions can be provided to reach that destination. Current traffic level and rides availability (from multiple means of transports) on the transport network is evaluated and several complex pattern matching mechanisms are put in place to shape the best multimodal way to reach the destination. Several approaches come from Operational Research [10, 4]. In [9], authors have considered ride-sharing as a complementary solution to usual means of transports in multimodal trip planning.

- **Chaperone Match.** Parents cannot bring their children to school every morning and they might find difficult to bring them back home when classes are over as well. When no other relatives or friends can look after a child, one can consider to share the path the child is going to follow, at a certain time, to look for someone that takes charge of assessing the presence of the child at intermediate checkpoints (e.g., a bus stop, a crossing, a public display, a store). Hardware sensors and reliable citizens located close the checkpoints can act as proximity probes and thus they can send actual feedback in real-time to the parents, and of course they send alerts when an unexpected event will occur.

The above examples in any case see a clear distinction between provider and requestor of a service, and consider that providers of a service are not influenced by the request. However, in a really integrated system, the mean to provide a service can be dynamically shaped upon the request, in a process of mutual influence. Indeed, those who provide a service is because they have a need to satisfy. It is thus possible to let the distinction between requestor and provider vanish, and dynamically adapt the shape of services depending on the need, also with some supra level objectives in mind behind the opportunistic self-interest of the involved parties.

# 3. INFRASTRUCTURAL REQUIREMENTS

Next generation smart mobility services should be pursued by settling some infrastructural requirements on its components. These requirements can determine the technical viability of smart mobility services deployment.

**Interconnection.** Based on the Internet of Things paradigm [2], the agents that populate the urban environment need to be connected and able to exchange messages each others. The distributed network of humans and ICT-devices will enable sensing, computing, and actuating capabilities only if information can flow seamlessly among a defined set of entities, despite network dynamics, and made ephemeral.

**Heterogeneity.** The inter-connected components of the ICT infrastructure are highly heterogeneous. This feature has not to be considered its weakness. We have to take advantage of their complementary role in knowledge mining. As example, one can consider a fixed entity on the roadside acting as a traffic sensor (e.g.: smart traffic light, CCTV camera). The data collected can be enriched with the one provided by mobile agents (e.g.: pedestrian, cars, buses), and hence its interpretation is made easier. Events detection and anticipation accuracy can improve as well.

**Interoperability.** Interoperable agents encourage combination of concurrent data streams from different locations, enabled in precise spatio-temporal patterns. Our coordination framework is based on the orchestration of such different data sources, dynamically selected due their complementary role, according to the incoming requests. Nevertheless, energy saving and classification accuracy should imply specific conditions that drive the concurrent activation of certain data sources and classifiers as well.

**Individual tasks.** Each agent has to share her knowledge among a collection of agents that provides complementary skills to her ones, in order to (i) "measure" the context of the surrounding environment, (ii) infer a certain situation, so become aware that is happening something relevant, (iii) and finally adapt the behavior of the actuators accordingly. Human actions and interactions are crucial during the whole process, and they can be tracked by explicit or implicit sensing of data through both personal devices like smart phones or smart vehicles, and through public interactive displays.

**Collective intelligence.** The brain of the system needs a software architecture designed by balancing a top-down and a bottom-up approach. The first usually results in very predictable and measurable systems that lack in reactivity in high dynamic contexts. The latter suits to cope with pervasive computing in decentralized systems, which their behavior is not always predictable, nor easy to be engineered. Collective intelligence can emerge from the reasoning and the collaborations among decentralized agents that aim to process individual and collective contents.

**System safety.** The system should own only a finite set of reachable states, which should be known in the design phase, and tested during the development. The aim here, is to avoid risks related to the eventual system's evolution towards uncontrolled situations. To enable this feature we need to own a deep understanding of system dynamics, and how to deal with them. In other worlds, citizens should feel safe to contribute in social collective intelligence initiatives, because they trust the system and its potentials, and find it useful in any circumstance.

**Information propagation.** The inferred information should pervade the nodes of the infrastructure till it can actually reach any potential agent that can be interested in it. Data should be packed in efficient structures, and routed via peer exchanges. A middleware architecture can be harnessed to reach these goals, and thus to support the purpose of the coordination framework, which is expected to become active supporter of agents interactions and facilitator of information propagation [7].

**Data management.** Big amount of spatio-temporally distributed and heterogeneous data will be concurrently evaluated by computing-enabled devices, at different stages. Thus, efficient storage, querying, and analysis practices are needed. Academic literature offers as many cues as many approaches it presents ([23], and [19] among the others), but a unified best practice is missing.

**Users privacy.** Discovering matches between needs and services implies computation on sensitive data coming from the set of agents. This task can contemplate the sharing of confidential information among them. Privacy concerns and sharing policies must be dealt on user agreements and should consider innovative practices to balance the value of the data shared with the value expressed by service enabled through the sharing of someone else [13]. One should be able to opt-out from collecting certain data once they could evaluate the purpose of that collection, the sharing rules, and the service(s) that could be enabled thanks to it.

# 4. CONCEPTUAL MODEL

As shown in Figure 3, the framework grounds on a matching engine that processes several data streams from a dense distributed tuple space, which is made of information concerning mobility status, requests, and services, generated by the agents on the mobility network. The rationale of the matching engine is triggered by incoming mobility requests, which in turns drive continuous processing steps. After several computing iterations on the available relevant information, the matching engine discovers and builds services on the mobility network, which finally result in mobility recommendations for the requesting user.

## 4.1 Distributed Tuple Space

Agents on the mobility network can implicitly or explicitly generate contextual information related to their mobility status, requests, and services. We believe a middleware infrastructure based on a set of networked tuple spaces [28] could represent a viable and suitable solution to store and share knowledge among all the agents interested in some particular generated contents, as well to properly feed the matching engine with the necessary information.

In particular, in the current demonstrative implementation of our infrastructure, we have built our coordination framework by exploiting the SAPERE tuple-based infras-

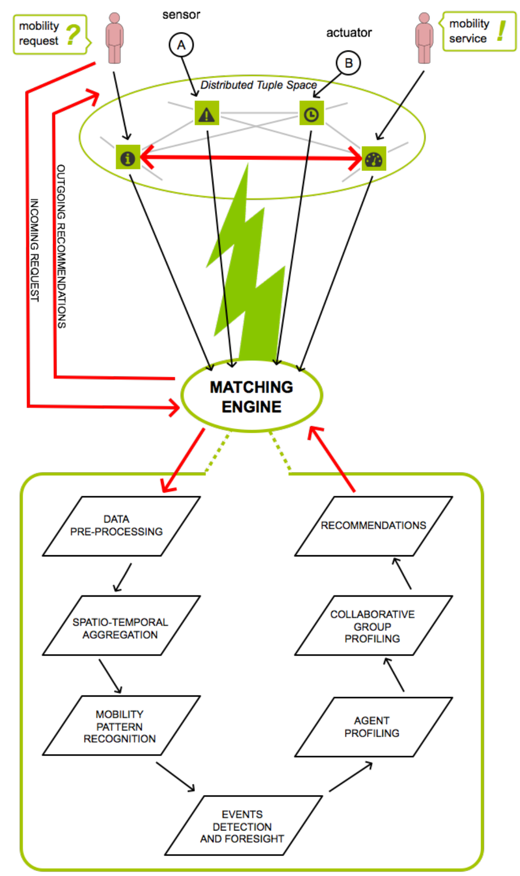



**Figure 3: Conceptualization of the coordination framework that matches mobility services with mobility requests. Agents on the mobility network include humans and ICT sensors-actuators such as smart traffic lights (which count the approaching vehicles) and smart signals (which change the displayed information). Data streams are dynamically selected and processed through the matching engine.**

tructure [31, 32]. SAPERE has the following characteristics that make them suitable to implement our proposed coordination infrastructure;

- It integrates an advanced and semantic pattern matching mechanism which can act as the basic building block to realize advance matches between mobility requests and offers;
- SAPERE defines a context-aware and spatial tuple space model, where one can adopt context-aware and spatial rules to dynamically select, evaluate, and propagate information, which is particularly suited to the area of mobility;
- SAPERE can associate specific middleware agents to react to events occurring in the network of tuple spaces, which can be used (and has been used, indeed) to realize advanced and multifold matching mechanisms, as described in the following subsection.

### 4.2 Matching Engine

The matching engine concept can be described through the definition of a set of sub-activities, each of which has been implemented as a SAPERE middleware agents. A description of the iterative phases that compose the matching engine follows.

**Data pre-processing and spatio-temporal aggregation.** At a first place, incoming data is filtered, cleaned and aggregated. The process of course needs a considerable amount of data, collected over time, until this activity results in meaningful content for the engine. At further iterations, each incoming raw data will be filtered, cleaned and aggregated again, according to spatio-temporal constraints of the incoming request.

Data modeling supports upper-level meaning abstractions, by generating complex data structures useful to understand a special mobility pattern of the considered agents.

**Mobility pattern recognition and events detection and foresight.** Machine learning techniques enable regular patterns identification and anomalies detection on the aggregated input data. Really well trained classifiers can perform effective anticipatory computing [24] that can be crucial in dynamic environments.

Not only agents on the move own mobility patterns (inferred, as example, by mining their mobility routes from GPS data). Roadside sensors can shape the mobility status on the mobility network as well, and so they let creation of tuples that characterize the mobility context of a geofenced area in a specific time interval. So, it is clear that it will be possible to detect and anticipate the occurrence of significant mobility events and have a real-time distributed representation of them.

**Agent and collaborative group profiling.** Each entity on the network is characterized by its own capabilities, which let it play specific activities with proper tasks, which are, in turns, driven by the nature of the entity itself. These conjectures bring the necessity to model agents behavior to the foreground (the Belief-Desire-Intention (BDI) model [26] is one of the approaches suggested by the literature in this field). The distributed tuple space should be populated with profiling contents related both to individual agents and to groups of them.

Interactions among group of agents is actually a crucial aspect to model. A survey with some proposals is presented in [6]. Humans interactions offers a good starting point in collaborative behavior understanding. The discovering of interaction reasons, modes, and effectiveness is pursued, in order to bring collaboration aspects to the shared knowledge. In order to motivate users in deeper collaborations, behavioral changes can be stimulated through tailored incentives and mechanisms taken from persuasion theory [11].

**Recommendations and feedback impacts.** Once the engine is able to infer the up-to-date context of the agents, the process goes on to the evaluation of which mobility events can be useful in addressing mobility requests. The set of identified alternatives is then sent to the requesting user, as recommendations, in the form of available services.

A similar mining can be performed when a reconfiguration of the ICT components on the mobility network is needed. Consider, as example, the increase of the sampling rate for a traffic sensor, according to the increase of variation in the traffic level measured. In that case, the granularity of the data collected should be increased. Hardware sensors and actuators have to be solicited with the optimal self-reconfiguration rules.

We believe the closing loop lets a profitable feature to come out from the coordination framework. It determines the continuous learning of the system, which becomes aware of how effective has been the mobility recommendations exchanged among the agents, and which benefits are generated thanks to them.

## 5. CASE STUDY EXAMPLE

To evaluate the effectiveness of the proposed coordination framework in the provision of smart mobility service, we have developed a set of algorithms that aims to reproduce the main conceptual activities involved in the matching engine described above.

We have focused our efforts on an Itinerary Match service, as described in Section 2. Our aim is to evaluate potential matches between mobility requests and offers. Commuters with a similar typical daily route should be detected and recommended to join ride-sharing opportunities. Of course, the framework should support the provisioning of integrated services, but our work is still on an initial stage and our testings have been delimited in shaping a single service.

Even if our framework is expected to collect real-time data, coming from the distributed tuple space, we have undertaken an offline experiment, by simulating the matching engine activities on a large dataset previously collected.

Raw data involved in our study covers one week of detections in the city of Turin, and it consists in Call Description Records (CDRs) collected by a mobile network operator, through the cellphone network. However, one can assume that data can be collected opportunistically from a set of drivers, through an application installed on their smartphones, and propagated on the nodes of the infrastructure.

Basically, each time a user performs data exchange on the Internet, starts a call, or sends a text message, a spatio-temporal record is created. Each occurrence contains the user's identifier (who makes it happen), the location of the antenna related to the network cell (where it has happened), and the timestamp (when it has happened).

### 5.1 Towards agents classification

According to the conceptual model of the coordination framework, the first step involves an initial **pre-processing of raw data**. In our case study, we have filtered data in a way that tries to exclude non-commuters. In particular, we define commuters as all the users that generate at least one event in both a pair of enough distant geographic zones (let us call them A and B), during working days. Furthermore, we have narrowed our definition of commuters by considering two particular regions to perform that filtering. We want to study urban mobility, so we have considered an area that covers the inner part of Turin as the zone A (about 100Km2 wide), and a geo-fence of a broader zone (about 3000Km2 wide), which surrounds the city center (suburban area), as the zone B. We have not made any consideration on the mean of transportation used by the users, because the input data is too fragmented and sparse. Best practice to succeed in this activity consists in excluding all the commuters that are used to move along railways, cycling paths, metro stations, or bus stops.

Next phase has involved the **spatio-temporal aggregation** of the selected CDRs into mobility traces.

We were interested in modeling data into upper-level meaning abstractions, useful to better understand mobility patterns of the considered users. We define a mobility trace as the conjunction of a pair of temporally adjacent events, which represents the origin-to-destination path covered by a certain user in a defined temporal interval. This process has resulted in the detection of sequential mobility traces (the destination of the first matches with the origin of the second) that can cover wide areas and time intervals.

Each user is characterized by a set of mobility traces that can be reduced in length by doing some further spatio-temporal aggregation. The aim here is to compress the amount of data linked to each user, by merging the mobility traces through their sequential relationships (in both spatial and temporal domains). Thus, they shape brand new, more extended, mobility traces. The amount of merging occurrences has also been stored in the resulting mobility trace. Spatial proximity has been computed on the pair of geographical points that characterize the origin or the destination on the pair of the involved traces. This is easy to compute with a point-to-point distance formula (e.g.: haversine, euclidean). Temporal closeness has been computed on the time interval associated to the same pair of mobility traces. This task is more tricky and it concerns the evaluation of several temporal relations. In our case we have followed the ones presented by Van Beek and Manchak [27].

For each user, the **mobility pattern recognition** phase has contemplated the inference of the most visited mobility path described by the mobility traces. In particular, this process has first resulted in the application of a K-means clustering algorithm on the spatial dimension of the mobility traces. The evaluation of the clustered points has been done on the amount of occurrences related to them. Only the two most populated clusters have been evaluated (origin and destination candidates). The typical daily route of each user has been discovered. Figure 4 shows an extract of the daily routes in a 1-hour time lapse.

We formally define a daily route as the most frequent outward plus the most frequent inward mobility traces generated by the same user from/to an origin to/from a destination. Actually, our daily routes mining has returned a significant result only for the 10% of the considered users, since most of the results have revealed the same amount of occurrences on multiple candidates in the same cluster (too much ambiguity on the data). We think this sudden loss of significance can be tackled by evaluating more temporally distributed data (one week of CDRs collection does not provide enough significance to our study).

The user's typical daily route can be useful **to detect and anticipate mobility events**. As examples:

- it describes the **daily journey** the user is used to perform, and so it represents a daily event itself;

- it reveals the **expected presence** of an agent on the underlying road network, during a certain time span;

- it can be harnessed to anticipate any **expected traffic congestion** on the underlying road network at a certain time;

- it lets to locate a moving probe that can be queried just in case in the future to detect **mobility status and alerts**.

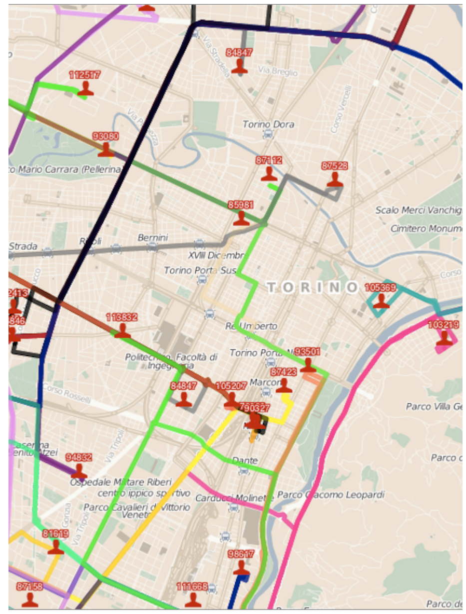



**Figure 4: Partial representation of the users daily routes in the city center of Turin at a given time interval.**

The contributions provided by each agent, wether it is a requesting agent or that it is a potential service provider, have been used to classify them, by creating their **agent profile** (in the distributed tuple space) with classifying labels and higher information contents.

## 5.2 Towards agents recommendation

Each agent profile is related to a commuter, and it initially contains only information about its daily route. In a real case scenario it can additionally include agent's personal details (such as demographics and interests) that can be collected through online social networks, its mobility preferences (such as its usual mean of transportation, and its willingness to do ride-sharing), the most likely home and work locations, its belonging to a same group of agents (due their commute similarities), and the most relevant historical events detected. Furthermore, one can think to add a ranking information to the agent profile that quantifies how much it has been involved in crowdsourcing and collaborative initiatives. This data can narrowly reflect social interactions among agents and their resulting benefits.Thus, it can outline the rise of **collaborative group of agents**.

As our next step, mining pool of users with similar daily routes has been done through an exhaustive search on all the users, by assuming that they were currently moving alone in a private car with 5 seats capacity. Each user should express at the same time its availability in offering ride-sharing services, and its necessity to find more efficient mobility solutions (in terms of vehicle occupancy rate). Our study has been limited to consider uniformed users that are characterized by the same mobility desiderata. However, on mining pool of users, one should consider maximum detour distance and time admitted as individual factors of the driver and each one of the passengers. A further improvement of the algorithm should contemplate the evaluation of the existing collaborative groups, in order to prefer users to rely on.

For any driver, we have selected the pair of mobility traces that composes its daily route, and we have compared each one of those with the whole set of concurrent mobility traces (within a confidence time interval) generated by other users. All the compared users with a mobility trace detected along the one generated by the selected driver represent potential passengers in ride-sharing pools, which can outline new potential collaborative groups of agents. The resulting pools contain information about the most suitable sequence of timed-stops to pick-up and drop-off passengers.

Once the pools are detected, ride-sharing **recommendations** can be sent out. Their should push social interactions between the involved users, and let new collaborative communities to be shaped. The system can track how they affect users mobility behavior and update both their individual and collaborative profiles.

## 6. RELATED WORK

Finding new approaches to enable mobility services has recently received a lot of attention. However, most of the studies are far from reaching effective and integrated solutions (from the collection of the requests to the provision of the services).

As discussed in Section 2, most of current ITS approaches do not offer a unified and integrated approach to support urban mobility in all its aspect, and often they own independent approaches for different mobility needs [1]. Also, in our proposed framework, and unlike most of ITS proposals, citizens are active agents of the overall infrastructure, by collaborating implicitly and explicitly towards the provisioning of smart mobility services.

Of particular interest to our work is the role of a middleware, which supports interactions and information exchange among the agents on the socio-technical system, and its involved in the generation of distributed intelligence. As far as we know, the best examples in this field that deal with the underlying infrastructure are the work of Harnie et al. [15], which aims to specify urban-area applications with tuple spaces abstraction, and the work of Julien and Roman [18], which proposes a middleware to enable context-aware mobile applications. The former enables intelligence through moving buses that carry the tuples, the latter propagates intelligence through vehicle-to-vehicle short range communications.

The works of Yang et al. and of Qu et al. [30, 25] introduce the concept of Intelligent Transportation Spaces as the integration of various ITS modules, vehicles, and roadside infrastructure. They mainly analyze safe and effective communication technologies to enable pervasive intelligence without impacting too much on drivers workload. However, neither of the works mention social interactions in matching mobility needs and services.

To the best of our knowledge, existing works do not give their contributions on proposing new approaches that could enhance social interactions.

Most of the mobility services presented in literature (e.g., [17, 12]) merely offer tailored solutions, without worrying about the creation of a coordinated methodology that deals with the dynamic orchestration of heterogeneous data streams.

We believe that a unified framework that models sensing, computing, and actuating capabilities of a socio-technical system of mobility agents is currently missing.

## 7. CONCLUSIONS AND FUTURE WORK

Social interactions among humans and ICT devices could strengthen the awareness of what urban mobility needs are, and how they can be addressed with smart mobility services. Social collective intelligence can be enabled, and so its utility can hit citizens and convince them to collaborate and cooperate each others through innovative sharing practices regulated by suitable incentives.

In the future, we will reshape our case study based on ride-sharing recommendations over a longer collection period, in order to reduce data ambiguity. Then, we will experiment with a larger set of mobility services, and will attempt at integrating them towards the realization of composite multimodal mobility services through our coordination framework.

## 8. ACKNOWLEDGEMENT

Work partially supported by the Emilia Romagna SPINNER2013 project MUCCA (Cooperative Urban Mobility in Smart Cities) and by the EU Project ASCENS (Autonomic Service Component Ensembles).

## 9. REFERENCES

[1] S.-H. An, B.-H. Lee, and D.-R. Shin. A survey of intelligent transportation systems. In D. Al-Dabass, Suwarno, J. Yunus, I. Saad, D. Giriantari, and A. Abraham, editors, *CICSyN*, pages 332–337. IEEE, 2011.

[2] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.

[3] N. Bicocchi, A. Cecaj, D. Fontana, M. Mamei, A. Sassi, and F. Zambonelli. Collective awareness for human-ict collaboration in smart cities. *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 0:3–8, 2013.

[4] J. Booth, P. Sistla, O. Wolfson, and I. F. Cruz. A data model for trip planning in multimodal transportation systems. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, pages 994–1005, New York, NY, USA, 2009. ACM.

[5] C. Brown, V. Nicosia, S. Scellato, A. Noulas, and C. Mascolo. Social and place-focused communities in location-based online social networks. *CoRR*, abs/1303.6460, 2013.

[6] G. Cabri, L. Leonardi, L. Ferrari, and F. Zambonelli. Role-based software agent interaction models: a survey. *Knowledge Eng. Review*, 25(4):397–419, 2010.

[7] G. Castelli, M. Mamei, and F. Zambonelli. The changing role of pervasive middleware: From discovery and orchestration to recommendation and planning. In *PerWare Workshop at the 9th IEEE International Conference on Pervasive Computing and Communications, Seattle (WAS)*, pages 214–219, March 2011.

[8] C.-W. Cho, Y.-H. Wu, C. Yen, and C.-Y. Chang. Passenger search by spatial index for ridesharing. *2012 Conference on Technologies and Applications of Artificial Intelligence*, 0:88–93, 2011.

[9] B. J. Coltin and M. Veloso. Towards ridesharing with passenger transfers. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1299–1300. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[10] B. de Jonge and R. H. Teunter. Optimizing itineraries in public transportation with walks between rides. *Transportation Research Part B: Methodological*, 55:212–226, 2013.

[11] B. Fogg. Persuasive computers: Perspectives and research directions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, pages 225–232, 1998.

[12] A.-J. Fougères, P. Canalda, T. Ecarot, A. Samaali, and L. Guglielmetti. A push service for carpooling. In *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, pages 685–691. IEEE, 2012.

[13] Z. Gao, M. Li, S. Du, and H. Zhu. Fairness-aware and privacy-preserving friend matching protocol in mobile social networks. *IEEE Transactions on Emerging Topics in Computing*, 1(1):192–200, 2013.

[14] D. Graziotin. An analysis of issues against the adoption of dynamic carpooling. *arXiv preprint arXiv:1306.0361*, 2013.

[15] D. Harnie, T. D'Hondt, E. G. Boix, and W. De Meuter. Programming urban-area applications for mobility services. *ACM Transactions on Autonomous and Adaptive Systems*, 9(2), 2016.

[16] W. He, D. Li, T. Zhang, L. An, M. Guo, and G. Chen. Mining regular routes from gps data for ridesharing recommendations. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, UrbComp '12, pages 79–86, 2012.

[17] X. Hu, V. Leung, et al. Vssa: a service-oriented vehicular social-networking platform for transportation efficiency. In *Proceedings of the second ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, pages 31–38. ACM, 2012.

[18] C. Julien and G.-C. Roman. Egospaces: Facilitating rapid development of context-aware mobile applications. *IEEE Transactions on Software Engineering*, 32(5):281–298, 2006.

[19] Y. Kwon, D. Nunley, J. Gardner, M. Balazinska, B. Howe, and S. Loebman. Scalable clustering algorithm for n-body simulations in a shared-nothing cluster. In M. Gertz and B. LudÃd'scher, editors, *Scientific and Statistical Database Management*, volume 6187 of *Lecture Notes in Computer Science*, pages 132–150. Springer, 2010.

[20] L. M. Martinez, G. Correia, and J. Viegas. An agent-based model to assess the impacts of introducing a shared-taxi system in lisbon (portugal). In *Proceedings of the 7th International Workshop on Agents in Traffic and Transportation*, 2012.

[21] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe. Parknet: Drive-by sensing of road-side parking statistics. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 123–136. ACM, 2010.

[22] S. Nawaz, C. Efstratiou, and C. Mascolo. Parksense: A smartphone based sensing system for on-street parking. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, MobiCom '13, pages 75–86, 2013.

[23] A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, and M. Stonebraker. A comparison of approaches to large-scale data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, SIGMOD '09, pages 165–178, 2009.

[24] V. Pejovic and M. Musolesi. Anticipatory Mobile Computing: A Survey of the State of the Art and Research Challenges, June 2013.

[25] F. Qu, F.-Y. Wang, and L. Yang. Intelligent transportation spaces: vehicles, traffic, communications, and beyond. *IEEE Communications Magazine*, 48(11):136–142, 2010.

[26] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1991.

[27] P. van Beek and D. W. Manchak. The Design and an Experimental Analysis of Algorithms for Temporal Reasoning. *Journal of Artificial Intelligence Research*, pages 1–18, 1996.

[28] M. Viroli, M. Casadei, S. Montagna, and F. Zambonelli. Spatial coordination of pervasive services through chemical-inspired tuple spaces. *TAAS*, 6(2):14, 2011.

[29] X. Yan, H. Zhang, and C. Wu. Research and development of intelligent transportation systems. In *Distributed Computing and Applications to Business, Engineering & Science (DCABES), 2012 11th International Symposium on*, pages 321–327. IEEE, 2012.

[30] L. Yang and F.-Y. Wang. Driving into intelligent spaces with pervasive communications. *IEEE Intelligent Systems*, 22(1):12–15, Jan. 2007.

[31] F. Zambonelli, G. Castelli, L. Ferrari, M. Mamei, A. Rosi, G. D. M. Serugendo, M. Risoldi, A.-E. Tchao, S. Dobson, G. Stevenson, J. Ye, E. Nardini, A. Omicini, S. Montagna, M. Viroli, A. Ferscha, S. Maschek, and B. Wally. Self-aware pervasive service ecosystems. *Procedia CS*, 7:197–199, 2011.

[32] F. Zambonelli, G. Castelli, M. Mamei, and A. Rosi. Programming self-organizing pervasive applications with sapere. In *Intelligent Distributed Computing VII - Proceedings of the 7th International Symposium on Intelligent Distributed Computing*, volume 511 of *Studies in Computational Intelligence*, pages 93–102. Springer, 2014.

[33] A. X. Zhang, A. Noulas, S. Scellato, and C. Mascolo. Hoodsquare: Modeling and recommending neighborhoods in location-based social networks. *CoRR*, abs/1308.3657, 2013.

# Negotiating Parking Spaces in Smart Cities

Claudia Di Napoli
Istituto di Calcolo e Reti ad Alte Prestazioni
C.N.R.
Naples - Italy
claudia.dinapoli@cnr.it

Dario Di Nocera*
Dipartimento di Matematica
University of Naples
"Federico II", Napoli - Italy
dario.dinocera@unina.it

Silvia Rossi
Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione
University of Naples
"Federico II", Napoli, Italy
silvia.rossi@unina.it

## ABSTRACT

Parking in urban areas is becoming a big concern for its environmental and economic implications. Smart parking systems are considered essential to improve both city life in terms of gas emission and air pollution, and motorists life by making it easier to park. Supporting technologies are emerging at the industrial level to easily locate available parking spaces, to automate parking payments, and to collect useful data on consumer demand. Nevertheless, the full potentiality of smart parking systems is still far to come, and it represents a big challenge for the future of Smart Cities. In this paper we propose to address the parking space allocation as the result of an agreement between parking providers and parking requestors that accommodates their respective requirements on some parking attributes. A software agent negotiation mechanism is adopted to establish such an agreement by taking into account user requirements on a parking space in terms of its location and cost, and the vendor requirements in terms of income and city regulations to obtain an efficient parking allocation and traffic redirection. It is shown that agent negotiation allows to allocate parking spaces to users in an automatic and intelligent manner by taking into account that a compromise among different preferences of users and vendors have to be met.

## Keywords

Agent negotiation, multi-agent systems, smart parking, smart cities.

## 1. INTRODUCTION

Urban transportation is considered a relevant investigation area for the innovation of Smart Cities since it may contribute to increase the quality of life of city-dwellers, to enhance the efficiency and competitiveness of the city economy, and to move towards the sustainability of cities by improving resource efficiency and meeting emission reduction targets. The main themes addressed in urban transportation are:

- Cooperative Intelligent Transport Systems and Services (C-ITS), based on the principle that all cooperative parties (i.e. ITS stations, vehicles, road side units) exchange information between each other, so enabling up-to-date traffic information, improved road safety and traffic efficiency.
- Enabling Seamless Multi-modality for End Users, based on the possibility to combine public transport with other motorized and non-motorized modes as well as with new concepts of vehicle ownership.
- Smart Organization of Traffic Flows and Logistics that involves multi-agency interaction, linking individual mobility with public transport services.

In this framework, one of the problems linked to the above themes, is parking in urban areas. It is widely recognized that drivers searching for parking in wide urban areas waste time and fuel, so increasing traffic congestion and air pollution [11]. Most of the research projects concerning smart parking systems focus on ways to collect and publish live parking information to drivers so they can be informed of available parking spaces near to the destination they require [9]. Nevertheless, the fragmentation of public and private parking providers, each one adopting their own technology to collect occupancy data, makes it difficult to advise motorists of available parking in multiple zones, but, more importantly, to help them in making decisions on where to park. Hence, smart parking applications should aim at coordinating individual parking solutions, both private and public, without involving end-users in the fragmentation of parking owners. Individual parking owners should be made aware of the benefits of such a global parking provision by showing them that the coordinated provision of parking solutions still guarantees their individual income and fair competition by better exploiting the parking spaces offered in a city.

In the present work, we investigate the possibility to use software agent negotiation to manage the relationship between parking supply and demand to provide user-oriented automatic parking services that take into account both drivers preferences, and parking vendors requirements together with social benefits for the city, such as a reduction of traffic by limiting parking in city center [13]. We propose to use software agents to model both a Parking Manager, who is responsible for coordinating the offers of individual Parking Owners (both public or private), and motorists who are end users that search for parking spaces that meet their requirements. In particular, an automatic negotiation mechanism

*Ph.D. scholarship funded by Media Motive S.r.l, POR Campania FSE 2007-2013.

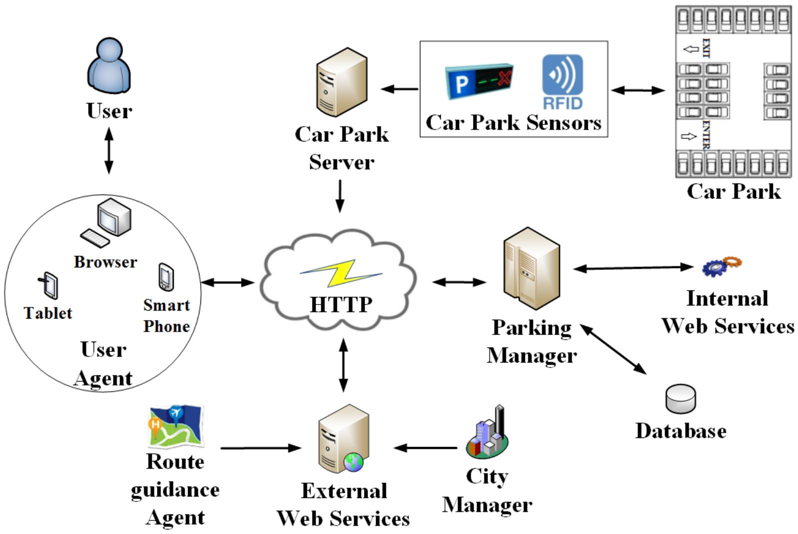



Figure 1: A Car Parking System.

is proposed to accommodate users and providers needs. Of course, the length of negotiation could prevent its use in this setting [5], so it should be adapted to the negotiation trend that may vary because of the the attributes to be negotiated upon, and the parking market situation.

## 2. A MODEL FOR A CITY PARKING SYSTEM

Car Park Systems refer to a wide spectrum of parking facilities including devices to automatically locate car parks and to automate parking space payment.

In the present work, a Car Park System is intended as a complex application composed of different devices and services, that allows users to retrieve information on the available parking spaces in a city around a specific destination area. A sketch of such a system is reported in Figure 1. As shown, a user may submit a request for a parking space to the Car Park Server through several devices (e.g. Tablet, Smart-Phone, PDA or PC). The system provides the user with a city map to select the area he/she would like to park, and an interface to indicate his/her parking preferences. A Parking Manager (PM) is responsible for processing the request. It queries an internal database (Database) to retrieve information on the available car parks, and it relies on specific applications to extract car park availability at the moment the request is processed (e.g. through Car Park Sensors). Also it may invoke additional services (External Web Services) to collect information on city regulations and/or events (provided under the responsibility of the City Manager) relevant to find a parking space, or other salient information, such as an estimation of the time necessary to arrive to the user destination from a specific car park, that can be retrieved from external applications as Google Maps API [10].

In such a framework, each car park is characterized by the following parameters:

```
car_park= <park_id, park_GPS_location, ref_price_unit,
          park_capacity, sector>
```

where `park_id` is the unique identifier of the car park, `park_GPS_location` is its GPS location, `ref_price_unit` is the default time unit price for a parking space, `park_capacity` is the total number of parking spaces of the car park, and `sector` represents the geographical location of the car park with respect to the city center. In fact, in the proposed application, the city is divided in several rings (referred to as *sectors*) that account for the distance between the car park and the city center, as shown in Figure 2. A `sector` is represented by an integer value so calculated:

$$sector = \begin{cases} 0 & distance\_from\_city \quad < min\_range \\ 1 + \left\lfloor log_2\left(\dfrac{distance\_from\_city}{min\_range}\right)\right\rfloor & otherwise \end{cases}$$

where $min\_range$ is the radius of the first area (`sector=0`), and $distance\_from\_city$ represents the distance between the car park location and the city center (located in `sector=0`).

A user request (`park_req`) is composed of values referred to the parking space attributes that are relevant for the user to decide where to park.

```
park_req(t)= <id_req, dest_GPS_location, start_time,
             end_time, reserv_time>
```

where `id_req` is the unique identifier of the user request, `dest_location` represents the GPS location of the destination the user wants to reach, the time interval (`end_time` - `start_time`) represents the duration the user wants to park for, and `reserv_time` is a flag used to distinguish between on-demand or advance requests. For the time being, only advance requests are considered since for on-demand requests different assumptions on the evaluation of car park occupancy should be considered.

With a static selection, the PM will select car parks considering only to meet the user requirements in terms of lo-

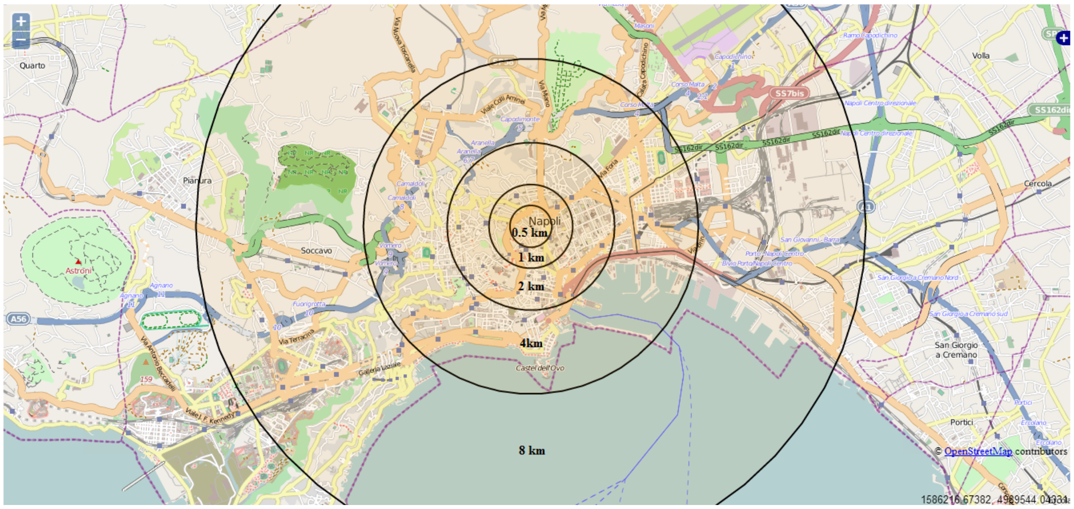



Figure 2: Sector distribution for the city of Naples.

cation, and available parking spaces for the required time interval. If there is no parking space meeting the requirements, a static mechanism will end up with no solutions for the driver request. A dynamic selection of parking spaces implies the evaluation of criteria that may not be explicitly expressed by the user, and that can influence the selection of the parking spaces offered by the PM. Furthermore, users may adopt private evaluation criteria that are specific to their profile to evaluate if the received offer is acceptable or not. With a dynamic selection, parking solutions that were not found with a static selection, could be produced as an acceptable compromise between PM and UA preferences.

## 3. NEGOTIATING OVER PARKING SPACE ATTRIBUTES

In a smart parking application, motorists will be classified according to their different requirements on parking spaces corresponding to different user's profile (e.g. business, tourist, generic). In fact, users may have different preferences on the parking attributes, and their relative *importance* (measured in terms of *weights*). Furthermore, additional information may be used (that could come from other sources of information) to help refining the selection process, e.g., unavailability of public transportation at the required time, the necessity to reach different locations once the car has been parked, the possibility to find other attractions in the area, and so on.

In this work, we investigate the possibility to use software agent negotiation to provide a user-oriented automatic parking service that takes into account both drivers preferences, and parking vendors requirements together with social benefits for the city. In particular, we propose a negotiation mechanism between two agents: the PM and a User Agent (UA). The PM has the aim to improve the citizen life, and city pollution by decreasing the influx of cars in the city center, and, at the same time, to offer a better distribution of vehicles in the managed car parks, still trying to obtain an economic income. The UA has the aim to help a motorist to select one of the parking solutions proposed by PM. Of course, it is difficult for the negotiating agent to evaluate whether to accept an offer to minimize the expected cost of communication (at the risk of getting a sub-optimal result for the specific application), or to keep on negotiating to maximize its expected utility (at the risk of increasing the cost of negotiation and ending with a conflict deal). Usually this lead to the specification of an acceptance condition that is not only based on utility, but on more complex criteria (i.e., based on utility and time) [2].

The adopted negotiation model is based on the one proposed in [6] that was shown to be a viable approach to address the problem of service selection for Service Based Applications characterized by Quality of Services values that once aggregated should meet user's preferences. The proposed mechanism allows to implement a *flexible* negotiation in terms of its length. In fact, the negotiation proceeds in *rounds*, and the number of round is not statically set, but its value may be changed by the PM or by the UA according to the trend of the negotiation process. A concession strategy is used at each negotiation round by the PM to make offers, and both negotiators may decide to end negotiation according to the negotiation evolution, so the negotiation deadline (i.e. the number of allowed rounds) is not fixed a priori.

### 3.1 A one-sided negotiation model

Usually negotiation takes place between two agents $x$ and $y$ willing to come to an agreement on conflicting interests, by exchanging an alternate succession of offers and counteroffers in a bilateral interaction [8].

In the present work we adopt the negotiation mechanism reported in [6], whose protocol is based on the Iterated Contract Net Protocol, that is frequently used to mime the human contract negotiation process [4]. Contract net protocol is a market-like mechanism allowing involved parties to exchange information in a distributed system, such as a multi-agent one.

As described in Figure 3, the protocol is organized in *negotiation rounds*, each one consisting of interactions between the UA, that is the *initiator* of the negotiation, and the PM, that is the agent proposing offers. Negotiation rounds may be iterated for a variable number of times until a *deadline* is reached or the negotiation is successful. Moreover, both the UA and the PM can stop the negotiation process. At each

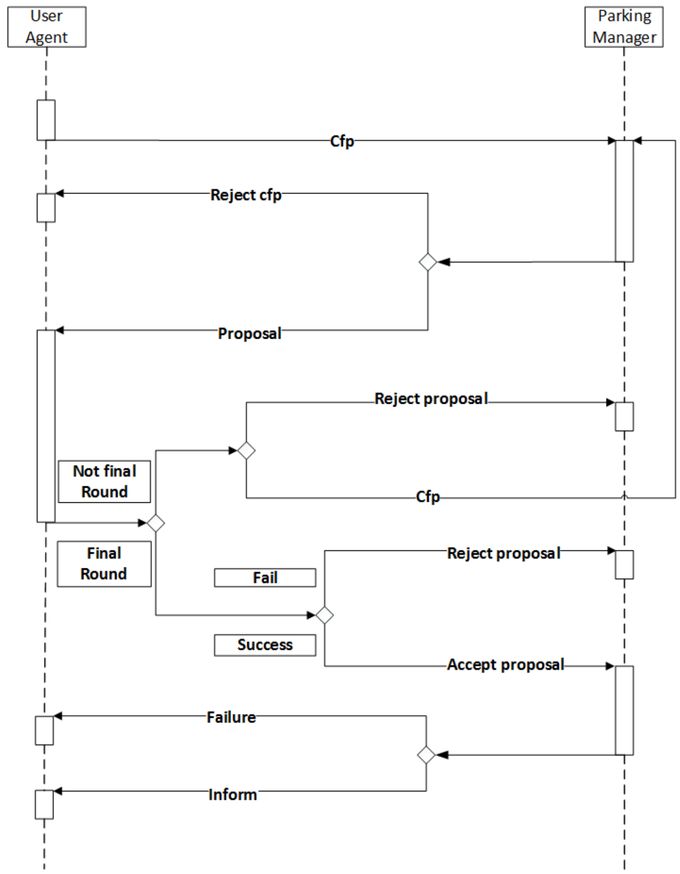



Figure 3: The iterated negotiation protocol.

negotiation round, the UA issues a request for a parking space (`cfp`) specifying its preferred values for the parking attributes; the PM can either reject the call (`Reject cfp`), if there are not offers available, or it sends back a parking solution (`Proposal`) selected from a set of available offers it calculated according to the preferences specified in the `cfp` and its own preference criteria. In the latter case, the UA evaluates the received offer, according to its own evaluation criteria, to decide whether to accept (`Accept proposal`) or to reject it (`Reject proposal`). If the offer is accepted the negotiation ends with an `Inform` message assigning the selected car space to the UA, otherwise a new round starts with the UA sending again the same `cfp` request. It should be noted that an offer proposed by the PM in a negotiation round is not considered available in future rounds once it is rejected. This assumption models the possibility that a rejected parking space may be offered to another user in the meantime, or its price may change according to the parking market trends.

Both PM and UA preferences over the attributes to be negotiated upon, are modeled through utility functions based on the Multi-Attribute Utility Theory defined on independent issues [3]. The function domain represents the *negotiation space*, and it is normalized to the interval $[0, 1]$. So, the utility function of an agent $x$ for an offer $o_y$ sent by the agent $y$ (with $x = y$ or $x \neq y$) is $U_x(o_y) : D_1 \times \cdots \times D^r \to [0, 1]$, where $D_1, \ldots, D_r$ are the value domains of the $r$ negotiation issues. The utility function allows to evaluate the value of each specific offer in terms of agent utility with respect to that offer.

In our model, the utility function of the PM depends on the car park availability at the moment the request is received, and on the distance of the car park from the city center, while the utility function of the UA depends on the parking space price, and on its distance from the requested destination. Different weights of the different issues may model different classes of UAs and PMs. In this way, the issues considered in the PM utility function take into account the preference of the PM to propose first car parks that are both less occupied and not located in the city center (to reduce the influx of cars in city centers). The issues considered in the UA utility function take into account the preference of the UA concerning the parking space price, and its location with respect to the preferred final destination. Utility functions are modeled as linear functions (as it will be explained in the following sections) resulting from the weighted sum of the considered issues.

The negotiation occurring between the PM and the UA is defined as a *one-sided* negotiation since it allows only the PM to formulate offers, according its own utility function, and the UA only to evaluate them, according its own utility function as well. The rationale of this choice is to model the assumption that UAs do not have complete information on parking spaces availability, otherwise they would simply choose the offer more convenient for them without reaching a compromise also with the preferences of the PM. So, at each round the PM sends only one offer (or equivalently a finite set of offers) selected according to a strategy allowing to take into account the requirements of both negotiators.

### 3.2 Parking Manager Behavior

At the first round of negotiation, the PM computes the set of possible offers corresponding to a set of car parks that meet the following requirements:

- the distance (referred to as `park_GPS_distance`) of the car park location (`park_GPS_location`) from the destination (`dest_GPS_location`) set by the user, is within a given distance (`location_tolerance`);
- the car park have spaces available for the time interval specified by the user at the time `t` the request is issued (`end_time - start_time`).

The `location_tolerance` is set by the PM in such a way to include also car parks that are not in the city center, and consequently they may be far from the `dest_GPS_location` specified by the user, since the PM tries to prevent users from parking in the city center and to maximize the occupancy of car parks.

An offer of the PM for a parking space of a selected car park is:

$$\texttt{offer(k)} = < \texttt{park\_id}, \texttt{park\_GPS\_distance}, \texttt{dest\_time\_distance}, \texttt{park\_price\_unit} >$$

where `park_id` is the identifier of the selected car park, `park_GPS_distance` is the distance between `park_GPS_location` and `dest_GPS_location`, `dest_time_distance` is the time necessary to travel from `park_GPS_location` to the `dest_GPS_location` using public transportation, and `park_price_unit` is the unit price offered for the selected parking space. The `dest_time_distance` value is obtained by invoking external services, such as Google Maps, but also also other city services giving additional information such as events preventing the use of public transport at the time of the request.

In order to incentivize users to park outside the city center and in car parks with more parking spaces available, the park unit price for a parking space is dynamically computed by considering that car parks located in the city center are more expensive (according to the ring distribution reported in Figure 1), and that car parks are offered with a discount factor that depends on the car park occupancy. Hence, the `park_price_unit` for a selected car park is computed as follows:

$$\texttt{park\_price\_unit} = \texttt{max\_price}\left(1 - \frac{\texttt{sector}}{\texttt{max\_sector + 1}}\right) + - \frac{\texttt{park\_availability}}{\texttt{park\_capacity}} \cdot u_d$$

where `max_price` is the maximum time unit price for the city center car parks, `max_sector` is the maximum number of sectors in the city, `park_availability` is the number of parking spaces available for the time interval requested by the UA (`end_time - start_time`), `park_capacity` is the total number of parking spaces, and $u_d$ is the maximum discount for the PM on the car parks (with $u_d \ll$ `max_price`). In this way, the price offered by the PM is not the static default price associated to the car park (i.e. `ref_unit_price`), but a dynamic value. The `park_availability` value is retrieved through a specific service invoked by the PM at the time the request is processed.

Once the PM computes the set of possible offers, it needs to establish which one to offer at each negotiation round, i.e. it needs to establish its concession strategy during negotiation. In order to do so, the PM uses a private utility function to rank the selected car parks. The evaluation function used by the PM to compute the utility of an offer ($offer_{PM}(k)$) is the following:

$$U_{PM}(offer_{PM}(k)) = \sum_{i=1}^{n}(\alpha_i * \frac{q_{i,k} - \min_j(q_{i,j})}{\max_j(q_{i,j}) - \min_j(q_{i,j})})$$

where $n$ is the number of issues the agent is evaluating, $q_{i,k}$ is the value of the i-th issue of the k-th car park, and $\min_j(q_{i,j})$ and $\max_j(q_{i,j})$ are respectively the minimum and the maximum values of the i-th issue among all the car parks selected by the PM. The constants $\alpha_i$ are weights associated to different issues with the constraint that:

$$\sum_{i=1}^{n}\alpha_i = 1$$

The issues for the PM are the distance of the car park from the city center, and the availability of parking spaces in the car park for the requested time interval, i.e.:

- $q_1 = dist(\texttt{park\_GPS\_location}, \texttt{center\_GPS\_location})$
- $q_2 = \texttt{park\_availability}$

Through its utility function, the PM ranks the offers for the selected car parks in a utility descending order (total or partial). At each negotiation round, it sends the UA one offer according this order, so adopting a concession strategy with a monotonically decreasing value of utility.

### 3.3 User Agent Behavior

The UA evaluates the offer it receives at each round to decide whether to accept or to reject it. In order to do so, it calculates its utility value for that specific offer, using the following utility function:

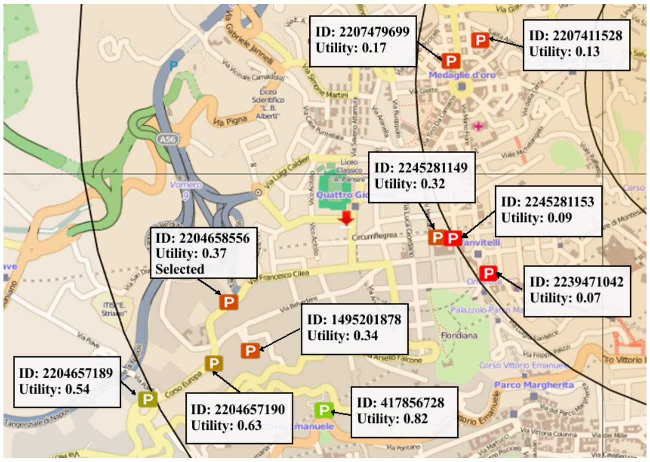
$$U_{UA}(offer_{PM}(k)) = 1 - \sum_{i=1}^{m}\beta_i * \frac{q_{i,k} - c_i}{h_i - c_i}$$

where $m$ is the number of issues the agent is evaluating, $q_{i,k}$ the value i-th issue of the k-th offer, $c_i$ is the preferred value over the i-th issue, and $h_i$ is a constant value introduced for normalizing each term of the formula into the set [0,1]. The constants $\beta_i$ are weights associates to different issues with the constraint that:

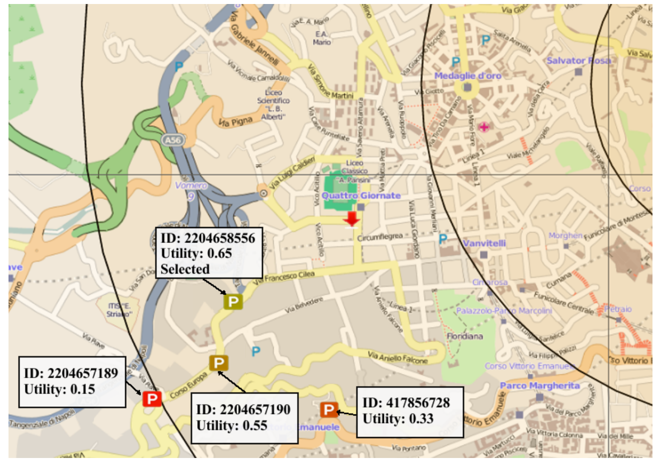
$$\sum_{i=1}^{n}\beta_i = 1$$

If $q_{i,k} - c_i < 0$ than the term $\sum_{i=1}^{m}\beta_i * \frac{q_{i,k}-c_i}{h_i-c_i}$ is set to zero. Moreover, we assume that the preferred $c_i$ values are not unreasonable with respect to each considered issue (i.e. user cannot ask for a parking space in a city center for free!).

The issues considered by the UA are the offered price, the distance of the offered car park from the requested location, and the travel time distance to the offered car park from the requested location with public transportation:




(a) Parking Manager utilities.




(b) User Agent utilities.

Figure 4: Parking Manager and User Agent Utilities.

- $q_1 =$ `park_price_unit`
- $q_2 =$ `park_GPS_distance`
- $q_3 =$ `park_time_distance`

The UA accepts the offer if the utility value for that offer is greater then a predefined *threshold* value. This threshold may be set to different values to model different UA profiles.

## 4. A FIRST EXPERIMENTATION ON A REAL SETTING

A preliminary set of experiments was carried out to determine whether negotiation is a viable approach in order to meet both users and parking managers requirements.

In this experimentation the weights in the utility functions are equally distributed among issues (i.e., $\alpha_i = 0,5$ and $\beta_i = 0,33$ for all $i$), while for each issue $i$, $h_i$ and $c_i$ are dynamically set to respectively $max_j(q_{i,j})$ and $mean_j(q_{i,j})$ (i.e., the maximum and the mean value for the current issue). The UA accepts an offer if its utility for that offer is greater than a threshold value set to 0.6 for all the experiments.

### 4.1 Utility Evaluation in a Running Example

A running example of a real negotiation, where we evaluate the utility obtained by the PM and the UA when an agreement is achieved, is reported.

The experiment starts with a request issued by a hypothetical user specifying the destination he/she wants to reach, selected on interactive city map provided by a specific service, and the time interval he/she wants to park for. As described in Section 2, the UA sends a `park_req` (i.e., a call for parking) to the PM. A graphical representation of the use case described above is reported in the Figure 4, where the destination selected by the user is identified with the down arrow.

At the first round, the PM selects a list of car parks around the user's destination (as shown in Figure 4(a)), and it calculates the ranking of the selected car parks based on its utility according to the function reported in Section 3.2. The PM found ten car parks with parking spaces available in the requested area within a predefined `location_tolerance`. Parking identifiers and locations are extracted from the OpenStreetMap database [7] of the city of Naples (Italy), while routing information (`dest_GPS_distance` and `dest_time_ distance`) are evaluated through the use of Google MAPs API [12]. The occupancy of car parks is randomly generated for each negotiation run. In the Figure 4(a), the selected car parks are reported with labels specifying the corresponding park ids and their utility values, as evaluated by the PM.

At each negotiation round, the PM offers to the UA the parking space with the highest utility value (in this example it offer a car park with utility equals to 0.82). The UA accepts (rejects) the offer if its utility for that offer, evaluated according to the formula described in Section 3.3, is higher (lower) than the threshold value. The first PM offer corresponds to an utility for the UA equals to 0.33. Hence, the offer is rejected because it is lower than the threshold value (equals to 0.6), and the UA starts another round of negotiation. The negotiation ends at the fourth round, when the UA accepts an offer with utility equals to 0.66 (corresponding to an utility for the PM equals to 0.37). In the Figure 4(b), car parks offered by the PM during negotiation are reported with labels specifying the corresponding park ids and their utility values, as evaluated by the UA.

In Table 1 we summarized all the relevant information at each negotiation round, reporting the number of parking spaces available in a car park (# Spaces), its distance from the city center (Distance), the unit price (Price) to be paid for the parking space, and the distance of the car park from the destination set by the UA, calculated both in length and in time (Route and Time), as obtained by querying a service of Google Maps. This information is necessary to allow the PM and the UA to calculate their utility values for the car parks, according to the utility functions reported respectively in 3.2 and 3.3. In this specific run, the negotiation ends after four rounds with an utility of the PM equals to 0.37 and for the UA equals to 0.66. Note that while the utility of the PM is not particularly high (because of the few parking spaces available in the car parks), the PM still manages to allocate a parking space in only four rounds of

| # Rounds | ID | # Spaces | Distance (m) | Price (€) | Route (m) | Time (s) | PM Utility | UA Utility |
|---|---|---|---|---|---|---|---|---|
| 1° | 417856728 | 109 | 3187 | 7.99 | 1516 | 1384 | 0.82 | 0.34 |
| 2° | 2204657189 | 41 | 4036 | 5.61 | 1818 | 2183 | 0.63 | 0.16 |
| 3° | 2204657190 | 41 | 3594 | 7.98 | 1192 | 871 | 0.54 | 0.55 |
| 4° | 2204658556 | 18 | 3359 | 7.46 | 891 | 646 | 0.37 | 0.66 |

Table 1: Negotiation on a single query.

negotiation, being able to reach a compromise by offering a car park that is not the closest to the user's destination, but still acceptable by the user in terms of time necessary to reach the destination from the car park location, and that is not too close to the city center.

## 4.2 1 vs N Rounds of Negotiation

Another experimentation was carried out on a simulation of 150 different queries made by users. The destinations selected by the user are located in sectors two and three on the city map. For each query a negotiation run takes place. The experimental results are summarized in Table 2 for successful negotiations. In particular, the table reports, for each negotiation run, the minimum, the maximum and the mean value (with the standard deviation) of the number of selected car parks (# Available car parks), the number of negotiation rounds (# Rounds), the PM and the UA utility.

The mean value of rounds (that is the mean number of offers sent by the PM) is much lower that the mean number of car parks selected by PM for the experiments (3.3 rounds with respect to 11 available car parks). This means that the negotiation ends before the PM offers all the selected car parks.

The obtained mean utilities values for the UA and PM are reported in rows 3 and 4 of Table 2, showing that a compromise on the requirements of both parties is reached. In fact, without negotiation (i.e., in the case the complete set of offers selected by the PM is known to the UA as well), the UA would select the offer that maximizes its own utility. The PM and the UA mean value utilities without negotiation are reported in the last two rows of Table 2. As expected, in this way, the UA requirements are privileged (UA achieves a mean utility value equals to 0.71) with respect to the PM ones (PM achieves a mean utility value equals to 0.35).

# 5. DISCUSSION AND CONCLUSIONS

Parking in populated urban areas is becoming a challenging problem requiring smart technologies in order to assist users in finding parking solutions, and to shorten the time necessary to find parking spaces. In this way, it is possible to decrease traffic congestion, and to improve the everyday life of city dwellers.

In the present work, we investigated the possibility to use software agent negotiation to address the parking problem by taking into account not only motorists' preferences regarding parking locations, but also parking vendors preferences regarding car park occupancy, and social city benefits (e.g. less traffic congestion in city centers). Multi-agent negotiation was already used in Intelligent Transportation System applications, such as [1, 4]. In particular, in [1] cooperative agent negotiation is used to optimize traffic management relying on shared knowledge between drivers and network operators about routing preferences. In [4] agent negotiation is used for dynamic parking allocation, focusing on satisfying driver's preferences on prices and distances.

Here we use a flexible negotiation mechanism to find parking solutions that represent a compromise among different needs: a user who prefers to park close to the city center, the car park vendors who prefer to sell parking spaces in less occupied car parks, and a city manager who tries to limit the circulation of cars in city centers. At this purpose, a Car Park System is proposed in order to provide a coordinated selling of parking spaces belonging to different car parks, managed by a single software entity, the Parking Manager agent. We show that an automated negotiation mechanism between the Parking Manager and motorists represented by User Agents, allows to find a compromise solution for the involved negotiators, through the use of utility functions that model different needs that have to be dynamically evaluated, so helping users in their decision making process. The automated negotiation mechanism allows to formulate offers that do not strictly meet the user requirements, and to find parking solutions that are a result of a negotiation process between the PM and the UA upon parking attributes that are evaluated differently by the negotiators.

In principle, the proposed framework allows also to model different user's profiles since the evaluation of the parking space attribute values may vary for different classes of users. Furthermore, different UAs and different PMs may adopt different evaluation criteria respectively to reject/accept and to select offers that can be based on dynamic parameters, e.g. as the occupancy of the car park at the requested time, or the unavailability of public transportation at the requested time.

Finally, we showed that negotiation is a viable and promising approach since a solution that is found before all selected car parks are proposed to users, i.e. before they reach complete information on the parking spaces available offers, and that does not privilege only the drivers' preferences.

In order to better assess the usability of negotiation in real parking settings, a further experimentation is planned to evaluate the length of the negotiation process when the number of car parks increases and their occupancy distribution varies because of multiple users' requests. Also, more experimental settings have to be designed with different values of the UA threshold, modeling the user's "attitude" to reach an agreement, to evaluate their impact on the negotiation length.

## Acknowledgements

The research leading to these results has received funding from the EU FP7-ICT-2012-8 under the MIDAS Project (Model and Inference Driven - Automated testing of Services architectures), Grant Agreement no. 318786, and the Italian Ministry of University and Research and EU under the PON OR.C.HE.S.T.R.A. project (ORganization of Cultural

| | max_value | min_value | mean_value |
|---|---|---|---|
| **# Available car parks** | 14 | 10 | $11 \pm 2$ |
| **# Rounds** | 9 | 1 | $3.3 \pm 2.5$ |
| **PM Utility** | 0.97 | 0.03 | $0.62 \pm 0.22$ |
| **UA Utility** | 0.75 | 0.10 | $0.68 \pm 0.06$ |
| **PM Utility 1 Round Neg** | | | $0.35 \pm 0.27$ |
| **UA Utility 1 Round Neg** | | | $0.71 \pm 0.04$ |

Table 2: Experimental Data collected in 150 runs.

HEritage for Smart Tourism and Real-time Accessibility).

## 6. REFERENCES


[1] J. L. Adler and V. J. Blue. A cooperative multi-agent transportation management and route guidance system. *Transportation Research Part C: Emerging Technologies*, 10(5âĂŞ6):433 – 454, 2002.

[2] T. Baarslag, K. Hindriks, and C. Jonker. Acceptance conditions in automated negotiation. In T. Ito, M. Zhang, V. Robu, and T. Matsuo, editors, *Complex Automated Negotiations: Theories, Models, and Software Competitions*, volume 435 of *Studies in Computational Intelligence*, pages 95–111. Springer Berlin Heidelberg, 2013.

[3] M. Barbuceanu and W.-K. Lo. Multi-attribute utility theoretic negotiation for electronic commerce. In *Agent-Mediated Electronic Commerce III, Current Issues in Agent-Based Electronic Commerce Systems*, pages 15–30. Springer-Verlag, 2001.

[4] S.-Y. Chou, S.-W. Lin, and C.-C. Li. Dynamic parking negotiation and guidance using an agent-based platform. *Expert Syst. Appl.*, 35(3):805–817, Oct. 2008.

[5] C. Di Napoli, D. Di Nocera, and S. Rossi. Evaluating negotiation cost for qos-aware service composition. In *Proceedings of the 14th Workshop "From Objects to Agents" co-located with the 13th Conference of the Italian Association for Artificial Intelligence (AI*IA 2013)*, volume 1099 of *WOA '13*, pages 54–59. CEUR workshop proceedings, 2013.

[6] C. Di Napoli, P. Pisa, and S. Rossi. Towards a dynamic negotiation mechanism for qos-aware service markets. In *Trends in Practical Applications of Agents and Multiagent Systems*, volume 221 of *Advances in Intelligent Systems and Computing*, pages 9–16. Springer International Publishing, 2013.

[7] M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE*, 7(4):12–18, 2008.

[8] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, methods and challenges. *Int. Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

[9] K. Nakamura, I. Hondo, N. Hataoka, and S. Horii. Car information systems for its. *Hitachi Review*, 49(3):102–106, 2000.

[10] B. Pan, J. Crotts, and B. Muller. Developing web-based tourist information tools using google map. In M. Sigala, L. Mich, and J. Murphy, editors, *Information and Communication Technologies in Tourism 2007*, pages 503–512. Springer Vienna, 2007.

[11] E. Polycarpou, L. Lambrinos, and E. Protopapadakis. Smart parking solutions for urban areas. In *2013 IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, 2013.

[12] G. Svennerberg. *Beginning Google Maps API 3.* Apress, 2010.

[13] D. Teodorović and P. Lučić. Intelligent parking systems. *European Journal of Operational Research*, 175(3):1666–1681, 2006.

# Capacity, Information and Minority Games in Public Transport

Paul Bouman† Leo Kroon† Gábor Maróti‡ Peter Vervest†
PBouman@rsm.nl LKroon@rsm.nl G.Maroti@vu.nl PVervest@rsm.nl

†Rotterdam School of Management, Erasmus University
Department of Technology & Operations Management
Burgemeester Oudlaan 50, 3062 PA, Rotterdam, The Netherlands

‡Netherlands Railways
Utrecht, The Netherlands
‡VU University, Amsterdam, The Netherlands

## ABSTRACT

Many public transport operators are faced with high peak demands. Often this leads to crowded vehicles and discomfort for the passengers. The increasing use of information technologies creates new opportunities for passengers to avoid crowding. However, the role of crowding in the dynamics of a public transport system is not well understood. With the definition and implementation of a model based on the minority games, a class of games that deals with crowding dynamics, we aim to provide public transport operators with insights to deal with crowded situations.

We propose an extension of a minority game where multiple resources and heterogeneous agent preferences are included. We have conducted two simulation studies, aimed at investigating the dynamics of crowding within public transport. In our first experiment we investigate the effect of the availability of information on crowding. In a second experiment we study the dynamic optimization of capacities according to a rolling stock circulation model. We find that both the availability of information disclosed and the chosen capacity optimization mechanism have an impact on the number of agents utilizing resources and their payoffs. As such, these models will allow us to develop new operator policies to deal with crowded situations in the future.

## Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Miscellaneous—*Coherence and coordination*

## General Terms

Experimentation, Management, Performance

## Keywords

capacity, coordination, information, minority games, public transport, resource allocation

## 1. INTRODUCTION

Operators in public transport are often faced with peak demands, typically during the morning and afternoon rush hours. As a result, vehicles can become very crowded, greatly reducing the comfort experienced by the passengers. As information technologies enable passengers to have more direct communications with the public transport operators and have more freedom to work at different locations, passengers are gaining more opportunities to avoid crowded situations. However, the impact of crowding on passenger behavior and the interaction between railway operations and passengers is not well understood. In this paper we develop a model, based on the concept of minority games, that allows us to study the dynamics of crowding in public transport through computational experiments and evaluate the impact of operational and behavioral models on a number of performance measures, most importantly the utilization of available capacities.

Since the "El-Farol Bar Game" [1] was first introduced in 1994, the concept of the *minority game* has received a lot of attention from researchers. One of the great strengths of this model lies in the simplicity of its description: a population of agents have to decide every Thursday night whether to go to the bar or not. Once they go the bar, they have a positive payoff if less than 60% of the population goes to the bar, while they have a negative payoff if it is too crowded. As everyone makes this choice every Thursday, the El-Farol Bar Game has an iterative nature. While historic information is provided, the interesting aspect comes from the fact that there is no direct coordination between the agents.

Issues related to limited availability of resources and a lack of explicit coordination occur in many real world systems. The applications of these models include car traffic [2], congestion in computer networks [8] and financial markets [4]. While these types of applications were considered earlier from a game theory perspective, most notably under the name of *congestion games* [11], the novelty from the "El-Farol Bar" study was the application of a complex systems approach enabled by simulation of a repeated game, while game theory is mostly concerned with the properties of equilibria.

In this paper, we focus on minority games where the operator cannot control agent behavior, but has control over the disclosure of information and the system capacities. The main application domain is public transport systems, where passengers share vehicles depending on their chosen route and time of travel. If a connection is operated frequently, passengers with some flexibility in their schedule can try to avoid crowded situations by shifting time of travel. Since it is reasonable that a passenger does not want to travel at any time, we introduce the concept of individual choice sets representing the acceptable choices. To our best knowledge, this type of heterogeneity of the choice sets has not been studied in the context of minority games before.

Within public transport systems, there are many opportunities to provide passengers with additional information: many stations and vehicles have screens with travel information, and many passengers use smart phones to receive information during their journeys. The increasing adoption of smart card ticketing systems allows operators to have accurate data on the utilization of each vehicle. As operators in railway and metro systems can extend or shorten the trains [5] and bus operators can employ different vehicle sizes, adaptive capacity allocation is becoming a possibility.

The main observation in the original "El-Farol Bar Game" simulations [1] is that even though individual agents keep switching their preferred predictive model, the aggregate utilization of the bar converges to the efficient level. In order to explain this phenomenon the minority game was introduced, where the utilization history was replaced with a history of binary values indicating whether the bar was overcrowded or not. The main idea of this approach is that the set of all possible deterministic strategies can be characterized so that methods from statistical mechanics can be applied [3].

The remainder of this paper is organized as follows: in Section 2 we introduce our class of minority games. In Section 3 we discuss the architecture of our simulation and agents. This simulation framework is then applied in order to investigate the effect of different information policies in Section 4. In a second simulation study we evaluated the effect of rolling stock optimization in the context of public transport (Section 5). In Section 6 we show that the inclusion of individual choice sets and scoring functions leads to NP-hardness of maximizing the efficiency of a given system. We discuss our findings and plans for future research in Section 7.

## Related Work

A variation of the minority games are the *resource allocation games*, introduced by [7]. This extension of minority games introduces multiple resources and capacities that vary over time. Conditions are given under which the agents can use a social network structure in order to adapt efficiently to variations of the capacities. The fluctuations of the capacities considered in the studies associated with the *resource allocation games* only depend on time and do not depend on the distribution of agents over the resources during the game.

While the body of knowledge on learning techniques for agents in minority games [10] is very useful for the engineering and design of artificial agents, it is a question whether it is applicable within systems where real humans are involved. Selten et al. [13] conducted a laboratory experiment involving route-choice. The participants could be divided into three groups: participants who had the tendency to switch away from a road if it was congested during the previous round, participants who had the tendency to stay on their current road regardless of it being congested during the previous round, and participants who were harder to classify. Although the participants showed different types of behavior, the distribution of the participants over the roads approached the equilibrium very closely.

## 2. A MODEL FOR CROWDING DYNAMICS

The general scheme of model is that in each round every agent decides whether he will use one or more resources or refrains from doing so. Using a resource gives the possibility to gain a positive payoff or a negative payoff depending on the utilizations encoutered. If he does not use any resource, the payoff will be neutral, i.e. zero.

We define symbols for the resources, the agents and payoffs. The resources will be defined in the following way:

- A set $\mathcal{R} := \{1, 2, \ldots, m\}$ of $m$ resources.
- A *soft* capacity function $\text{cap} : \mathcal{R} \to \mathbb{Z}^+$.

Thus there are $m$ resources, each of which having an associated capacity. Note that we define *soft* capacities: they can be violated, but everyone in such a situation should have a negative payoff. Based on the capacity we define the utilization of a resource as the fraction of its capacity that is occupied. The typical example in public transport is the number of passengers divided by the number of seats. As the game is played iteratively, the transport operator can adapt the capacities based on observations recorded during earlier rounds of the game. We also define the preferences and payoffs of the agents that play the game:

- A set $N = \{1, 2, \ldots, n\}$ of $n$ agents.
- A non-empty collection $C_i$ of subsets of $\mathcal{R}$.
- A scoring function $s_i : \mathbb{Q}^m \times C_i \to \mathbb{R}$ for each agent $i \in N$

During each round, every agent should choose one of the options in its choice set. We assume that every choice set contains the empty set as a neutral option, but this is not strictly necessary. We can describe the outcome of a round based on the choices made by all agents. If an agent $i$ chooses to use a set of resources $c \in C_i$, we set the indicator variable $x_{ic}$ to 1. The set of all vectors of $x_{ic}$'s describing a valid outcome is thus defined by

$$\mathcal{O} = \{x \,|\, \forall i \in N : \sum_{c \in C_i} x_{ic} = 1, x_{ic} \in \{0, 1\}\}. \tag{1}$$

Given the outcome vector $x$ for a round, we can calculate the utilization of the resources. We define a vector $u(x) \in \mathbb{Q}^m$ that contains an entry for each resource. The entry $u_r(x)$ for resource $r \in \mathcal{R}$ is calculated as follows:

$$u_r(x) = \frac{\sum_{i \in N} \sum_{c \in C_i : r \in c} x_{ic}}{\text{cap}(r)}. \tag{2}$$

While in principle $s_i$ can be a general scoring function, for ease of analysis we will use the restricted class of *threshold based scoring functions*. These scoring functions have a payoff of $-1$, 1 or 0 depending on a individual threshold $\theta_i$ and the maximum encountered utilization. The scoring function itself is then defined as follows:

$$s_i(u, c) = \begin{cases} 0 & \text{if } c = \emptyset, \\ 1 & \text{if } \max_{r \in c} u_r \le \theta_i, \\ -1 & \text{otherwise.} \end{cases} \tag{3}$$

## Performance Measures

Since we want to analyze the behavior of an agent population, we will introduce some measures that are of analytic interest and can be recorded during a simulation. We define the # symbol to denote the cardinality of a set (e.g. $\#\{6, 9\} = 2$). Given an outcome $x \in \mathcal{O}$ during any of the rounds of the game, we can calculate the following observations:

- $\text{utl}(x) = \frac{1}{n}\#\{i \in N : x_{ic} = 1, c \neq \emptyset\}$, i.e. the fraction of agents utilizing a resource.
- $\text{pos}(x) = \frac{1}{n}\#\{i \in N : x_{ic} = 1, s_i(y, c) > 0, y = u(x)\}$, i.e. the fraction of agents with a positive payoff.
- $\text{posc}(x) = \frac{\text{pos}(x)}{\text{utl}(x)}$ is the fraction of agents with a positive payoff among the agents who utilize a resource.
- $\text{avg}(x) = \frac{1}{n}\sum_{i \in N}\sum_{c \in C_i} s_i(u(x), c)x_{ic}$ is the average payoff of the agents.

## 3. ARCHITECTURE OF THE AGENTS AND SIMULATION

Given an instance of the game, a simulation still depends on two more aspects: the way the agents make their decisions and to which extent the agents can observe the outcome of the previous rounds. As we want to be able to evaluate the effect of different types of agent behavior, we will allow different types of agents in the population. We will introduce a number of types in Section 3.1. We first define the main steps that will be executed in each round of the simulation:

1. Let every agent $i \in N$ choose one option $c \in C_i$ from its choice set according to its agent type.
2. Calculate the outcome vector $x$ and corresponding utilization vector $u(x)$ accordingly.
3. Let every agent $i \in N$ observe, learn and process its score $s_i(u(x), c)$ based on its agent type.
4. The operators lets every agent $i \in N$ observe, learn and process information based on the active information policy and the utilization vector $u$.

From these steps we can see the necessary ingredients for an agent implementation within this simulation scheme: an agent needs a choice function and can optionally implement a method to process incoming scores and information.

### 3.1 Agent Types

The most simple agent type is the *random agent*, who selects a choice from its choice set uniformly at random in each round. This agent type is useful for both benchmarking purposes, validating the simulation architecture analytically, and to model noisy behavior within the population.

The more complicated agent types will make decision based on observations during earlier rounds of the game. For these agent types, step 4 of the simulation process in a round can have an effect on step 1 in the next round. The number of rounds the agents look back is referred to as the *memory length*. An important finding in the minority game model is that the most efficient utilization is reached when the memory length of the agents is proportional to the logarithm of the total number of agents [12].

The second type of agent, the *average payoff agent*, applies a simple reinforcement learning heuristic. Reinforcement learning strategies have received notable attention in the literature, and we take one of the most simple ones as an example. As such, our *average payoff* agents perform exploration during 10% of the rounds by making a random choice, while they exploit the observed average payoff values during 90% of the rounds. In case multiple choices have the best average payoff, the tie is broken by picking one option uniformly at random.

A variation of average payoff agent is the *average utilization agent*, who uses the same reinforcement learning heuristic to learn the average utilization of the resources. The main difference is that this agent uses the information received to learn the average utilization and pick the choice with the lowest average utilization, or the neutral option if this choice has still higher average utilization than its threshold.

The last type of agent, the *predictive agent*, aims to predict future utilizations in order to find the best choice. If the agent can predict future utilizations, the agent can generate a fictitious utilization vector and evaluate the expected score of each choice. This agent type is similar to the one studied in original El Farol Bar paper [1]. In a round with index $t$, the agent checks which of its personal heuristics was most accurate in round $t-1$ and uses this one to predict utilizations in round $t$. As our model introduces the concept of multiple resources, there can be situations where an agent does not know all historic utilizations of each resource. We calculate the accuracy of each heuristic based only on the information that is available. As availability of utilization information is defined on the agent level, an agent can compare the heuristics using the same data set.

We implemented the following predictive heuristics: replicate the oldest utilization in memory, take the average of the utilizations in memory or fit a linear regressive model on the utilizations in memory.

### 3.2 Information Policies

At the end of each simulation round, we let each agent process information and observations on the utilization of the resources. We define a unit of information as a 3-tuple $(t, r, u)$ consisting of the round of the game $t$, a resource $r$ and a utilization vector $u$. As the agent can have multiple resources in its choice set, it should be able to receive and process multiple pieces of information each round. In general, an *information policy* is a set of rules that determine the information offered to each agent in each round. While there are very many information policies possible, we propose four basic ones.

In public transport, the fact that an agent is using a resource allows it to observe the utilization. Thus in our most basic information policy, *private information*, an agent receives exact information for the resources in its choice.

On top of private information, the entity or agent controlling the resources could monitor the utilizations and try to attract more agents in case a resource $r$ has a low utilization, say less than 40% of its capacity. In such a situation the information policy can state that additional information regarding resource $r$ should be provided to all agents. We will refer to this type of policy as *adaptive information*.

In some situations there are information systems that provide information on the crowding of a resource. A real life example one can think of is a smart phone application of a

Table 1: Results of the simulation study where different information policies are evaluated. The minimum, average and maximum utl (fraction of agents utilizing a resource) and posc (fraction of agents who have a positive payoff among those that utilize a resource) values measured for each of the 66 population mixtures are reported.

| utl ($n = 50$) | **private** | **adaptive** | **estimate** | **full** |
|---|---|---|---|---|
| Minimum | 0.63 | 0.75 | 0.75 | 0.75 |
| Average | 0.77 | 0.90 | 0.91 | 0.91 |
| Maximum | 0.91 | 0.97 | 0.99 | 0.99 |

| utl ($n = 100$) | **private** | **adaptive** | **estimate** | **full** |
|---|---|---|---|---|
| Minimum | 0.45 | 0.66 | 0.70 | 0.73 |
| Average | 0.60 | 0.70 | 0.73 | 0.77 |
| Maximum | 0.75 | 0.75 | 0.82 | 0.85 |

| utl ($n = 200$) | **private** | **adaptive** | **estimate** | **full** |
|---|---|---|---|---|
| Minimum | 0.20 | 0.40 | 0.46 | 0.51 |
| Average | 0.42 | 0.48 | 0.52 | 0.56 |
| Maximum | 0.75 | 0.75 | 0.75 | 0.78 |

| posc($n = 50$) | **private** | **adaptive** | **estimate** | **full** |
|---|---|---|---|---|
| Minimum | 0.90 | 0.90 | 0.89 | 0.87 |
| Average | 0.95 | 0.93 | 0.91 | 0.90 |
| Maximum | 0.99 | 0.97 | 0.93 | 0.92 |

| posc ($n = 100$) | **private** | **adaptive** | **estimate** | **full** |
|---|---|---|---|---|
| Minimum | 0.46 | 0.46 | 0.41 | 0.30 |
| Average | 0.77 | 0.65 | 0.60 | 0.48 |
| Maximum | 0.98 | 0.84 | 0.72 | 0.54 |

| posc ($n = 200$) | **private** | **adaptive** | **estimate** | **full** |
|---|---|---|---|---|
| Minimum | 0.02 | 0.02 | 0.02 | 0.02 |
| Average | 0.49 | 0.35 | 0.27 | 0.19 |
| Maximum | 0.94 | 0.68 | 0.42 | 0.26 |

public transport operator, that shows one, two or three icons based on the forecasted crowdedness of a vehicle, reducing the utilization level provided to the agent to a few discrete values. This idea is capture by the *estimate information policy*, where the utilization of each resource is rounded up to either 0, $\frac{1}{4}$, $\frac{2}{4}$, and so on, similar to the 3 symbol crowding indicators provided by some operators. This rounded utilization is then provided to all agents. We should take care that we send the rounded utilizations in case the agent did not observe the utilization by itself, and use exact utilization otherwise.

In the final template, we send out exact information on every resource to every agent in each round – thus in this situation the agents have *full information*.

# 4. EVALUATING INFORMATION POLICIES

In our first experiment, we evaluate the four information policies in a population of agents that use public transport to travel from a single origin to a single destination, but can choose for different times of travel. As such their choice sets contain only singleton resources, reflecting the departure times a public transport service is scheduled and the empty set as a neutral option, reflecting a journey by car or staying at home. We find that increasing the available information leads to a greater number of agents utilizing the public transport system, but at the cost of the average payoff. However, the magnitude of this effect is influenced by the ratio of population size and available capacity.

## 4.1 Experimental Setup

In our experiments, we work with $m = 10$ resources representing the departure times. Every choice set $C_i$ contains $\emptyset$ and 3 different singleton sets picked uniformly at random from $\mathcal{R}$ without replacement. For each agent $i$ we use a threshold based scoring function with $\theta_i \in \{\frac{5}{10}, \frac{6}{10}, \ldots, 1\}$ picked uniformly. The capacity of each resource is fixed to 10, i.e. $\text{cap}(r) = 10$.

As we have 100 units of capacity available each round, we consider a high capacity scenario with $n = 50$ agents, a regular scenario with $n = 100$ agents and a low capacity scenario with $n = 200$ agents. For each of these scenarios, we vary the population by picking all pairs $p, q \in 0, 1, \ldots, 10$ such that $p + q \leq 10$. Our population then consists of $10p$ *random* agents, $10q$ *average utilization* agents and $10(10 - p - q)$ *predictive* agents. In total 66 population mixtures are evaluated. For each mixture of agent implementations we regenerate the choice sets and thresholds 100 times. For a given instance of the choice sets we run the experiment 25 times, regenerating the predictive agents 5 times if they are part of the population. Thus, in total we run 2500 simulations per combination of population mixture and population size. As we want to ignore the warm-up period of the simulation and like to interpret the rounds as days, the measures are recorded from round 10 to 40 during each simulation run.

The predictive agents each have an individual randomly selected set of 3 random predictive heuristics from the following list: average heuristic with memory lengths of either 4, 5, 6 or unlimited, linear regression with memory lengths of 4, 5, 6 or unlimited, replicate the oldest observation with memory length either 1, 2 or 3.

## 4.2 Results and Discussion

The results of our simulation experiments are presented in Table 1. If we look at the left column of Table 1, we can verify that when we increase the level of information provided to the agents, the number of agents utilizing a resource increases. If we look at the average values from private to adaptive, we can see that the 0.13 increase for $n = 50$ scenario is greater than the 0.06 increase for $n = 200$. These numbers suggest that the effect of information depends on the units of capacity available per agent in the population.

If we look at the fraction of agents utilizing a resource with a positive payoff (this can be interpreted as customer satisfaction) in the right column of Table 1, we can see that increasing the level of information decreases the posc value. This seems intuitive, since adding information attracts more agents, and having more agents increases the likelyhood of crowding. Again, the amount to which the posc value decreases when we move from the *private* to the *adaptive* case is impacted by the amount of capacity available per agent: for $n = 50$ the decrease of 0.02 is less dramatic than the 0.14 decrease of in the $n = 200$ case.

For future work we will investigate whether better information policies can be designed. The are also questions regarding the effect of noise in communications, such as technical problems at the side of the operator or agents ignoring information sometimes. We are curious to learn whether such noise could lead to less correlated agent behavior and whether this can lead to better system efficiency.

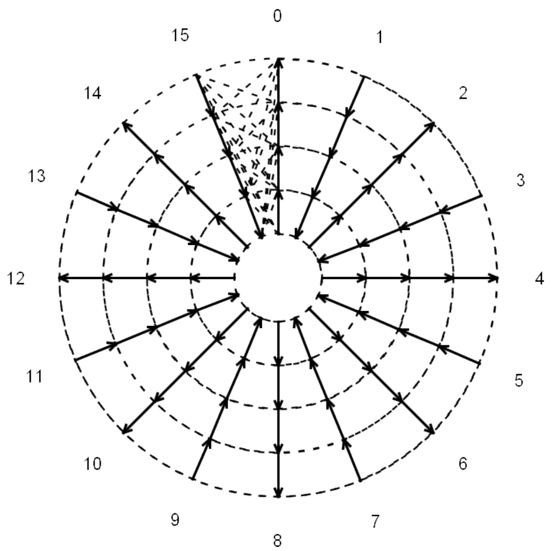



Figure 1: The network of trips along a line during the 16 time slots with the possible movements of rolling stock units outside the regular trips represented by dashed lines.

## 5. CAPACITY OPTIMIZATION IN PUBLIC TRANSPORT

In our second experiment, we want to evaluate the effect of rescheduling capacities on the crowding dynamics in the system. Consider a public transport scenario where a train moves back and forth a line of 5 stops. The train drives 8 full cycles per day and as moving along the line in one direction gives us 4 trips between the stops, the timetable consists of $4 \cdot 2 \cdot 8 = 64$ trips are offered each day. As individual travelers want to travel between two stops that are not necessarily connected by a single trip, a journey can consist of one or multiple trips. We will assume passengers always want to travel in the direction of their destination and as such for each origin-destination (OD) pair there are 8 different time slots at which passengers can make their journeys.

In order to facilitate the flow of passengers, the trains need to be long enough in order to allow comfortable transportation. To achieve this, the operator monitors utilization of vehicles and adapts the assigned number of rolling stock units to each train accordingly. The operator can decide how often the observed utilizations are evaluated to build a new rolling stock model. In this experiment, we will assume that this will happen periodically. The number of rounds after which the operator produces a new rolling stock schedule will be referred to as the *reschedule period*, denoted by an integer $k$.

### 5.1 Capacity Allocation

As the use of rolling stock units determines a significant amount of the operational costs of a public transport operator, they try to monitor the utilization of the train vehicles and adapt the capacities if necessary. The typical model used to determine the rolling stock allocation in these situations is by constructing the network of possible train movements, specifying a minimum demand on the arcs that correspond to passenger trips and look for a minimum cost circulation [6] based on operational costs.

We implemented a module in the simulation that represents an operator which dynamically optimizes demand. During each round of the simulation, train utilization for each trip is recorded. After $k$ rounds, the demand of a trip is set to $\mu + 2\sigma$, where $\mu$ is the mean utilization and $\sigma$ its standard deviation during those $k$ rounds. We chose this rule because similar rules are employed by real operators. The capacities of each trip are then calculated according to a rolling stock circulation, where we define a cost of 1000 per unit used and a costs of 1 for moving a unit between consecutive stops on the line. These numbers represent that buying and maintaining rolling stock units is a lot more costly than moving them around. We also assume that storing a unit at a station does not impose any costs. As a result, the minimum cost rolling stock circulation will minimize the number of units required before minimizing the movement costs, given that the defined demand must be met.

We use a minimum cost circulation algorithm [6] (which shares quite a lot of similarity with the well known augmenting path methods for max flow) to obtain the capacities. Although more efficient algorithms exist for this problem, the augmenting path method is straightforward to implement and fast enough for our simulations. The input network is visualized in Figure 1. The straight arcs represent movements between the stops and must carry the determined demands. The circular arcs represent storing a vehicle at a stop. The overnight arcs represent the purchase costs of the vehicles and the overnight balancing movements.

While the algorithms employed by operators need to take many different types of rolling stock and regulations into account [5], for reasons of simplicity and interpretability we assume that we have only one type of rolling stock with a nominal capacity of 10 seats.

### 5.2 Experimental Setup

In order to set up the simulation, we define a resource set that consists of the trips, so based on the 5 stops and 16 timeslots, we get $m = 64$ resources. The choice set of an individual agent is generated as follows: we pick two stops $o \neq d$ from among the five stops. By choosing $o$ as the origin and $d$ as the destination, the direction along the line is defined. We then pick 3 from the 8 available time slots corresponding to this direction in order to define the acceptable journeys. The choice set then consists of the empty set and the sets of trips corresponding to the journeys drawn randomly. Again we work with threshold based scoring functions where the threshold is picked uniformly from $\{\frac{5}{10}, \frac{6}{10}, \ldots, 1\}$.

For the purpose of simplicity, we use only one type of agent during this experiment: the *average payoff agent*. One of the reasons to choose this agent implementation is that software packages for dynamic traffic equilibrium computations with feedback use this approach. We pick the number of agents simulated as $n = 1000$. The reason to take a relatively large agent population is because we have 64 trips and as each trip should have at least one unit of rolling stock available, the available capacity is at least 640. In order to have a high probability to facilitate all the demand during the first rounds of the simulation, we set the initial demand of rolling stock units for each trip to 5. We also checked initial rolling stock counts of 1 and 10 units and our findings were robust under these variations.

Our goal is to evaluate the effect of different rescheduling periods. As the demand observed depends on the length of

Table 2: Results of the simulation study where the effect of rolling stock optimization on average payoff, operator costs and rolling stock units required is evaluated. The measures at round 100 of each simulation are reported, for different rescheduling periods (k) of 1, 5 and 10.

| $k = 1$ | min | mean | (±std.) | max |
|---|---|---|---|---|
| utl | 0.38 | 0.42 | (±0.02) | 0.46 |
| posc | 0.60 | 0.73 | (±0.05) | 0.84 |
| avg | 0.08 | 0.19 | (±0.04) | 0.28 |
| cost | 2092 | 2532 | (±498.5) | 3112 |
| units | 2 | 2.43 | (±0.497) | 3 |

| $k = 5$ | min | mean | (±std.) | max |
|---|---|---|---|---|
| utl | 0.58 | 0.60 | (±0.015) | 0.66 |
| posc | 0.74 | 0.85 | (±0.03) | 0.92 |
| avg | 0.28 | 0.43 | (±0.04) | 0.50 |
| cost | 3166 | 4120 | (±314) | 4120 |
| units | 3 | 3.9 | (±0.31) | 5 |

| $k = 10$ | min | mean | (±std.) | max |
|---|---|---|---|---|
| utl | 0.62 | 0.66 | (±0.01) | 0.70 |
| posc | 0.76 | 0.86 | (±0.03) | 0.92 |
| avg | 0.36 | 0.48 | (±0.04) | 0.56 |
| cost | 4188 | 4219 | (±142) | 5208 |
| units | 4 | 4.02 | (±0.14) | 5 |

the rescheduling period, we evaluated periods of 1 round, 5 rounds and 10 rounds. For each of these rescheduling policies, we generate 50 agent populations of choice sets with corresponding thresholds. For each population we then run 2 simulations of 50 rounds.

## 5.3 Results and Discussion

The time series distributions of the observed values of utl and posc during the 100 simulation runs for each of the three policies are presented in Figure 2. The distributions of average payoffs, the operational costs (according to the minimum cost rolling stock circulation) and the number of rolling stock units utilized during the last round of the simulation are presented in Table 2.

As we increase the length of the period, we can observe that utl increases. If the reschedule period is 1 round, we can observe from Figure 2a that it converges to a mean of 0.42. For a reschedule period of 5 rounds it converges to a mean of 0.6 (Figure 2c) and for a reschedule period of 10 rounds it converges to a mean of 0.66 (Figure 2e). One possible explanation for the fact that longer periods give a higher value for utl is that a longer reschedule period has the potential to yield a more stable mean and possibly more accurate standard deviation (except for the case where the period is 1; then the standard deviation is always 0). As the mean and standard deviation have direct effects on the demand and thus the capacities that are calculated, they seem likely causes for the observed behavior.

For the fraction of agents that utilize a resource and have a positive payoff posc, we can observe that it converges to a value of 0.73 for a reschedule period of 1 round (Figure 2b), to a value of 0.85 for a reschedule period of 5 rounds (Figure 2d) and to a value of 0.86 for a reschedule period of 10 rounds (Figure 2f). The 1 round scenario has a higher standard deviation than the other scenarios .

While both the utl and posc have higher averages for longer reschedule periods, Figure 2 also shows slower convergence for longer reschedule periods. Table 2 also suggests that longer reschedule periods lead to higher costs and a higher number of rolling stock units required. However, this can be explained by the increase of the utl value. A final interesting observation in Table 2 is that the average payoff for the agents also increases if we use longer periods.

Our results suggest that there are many disadvantages for the single round reschedule period. Increasing the period may lead to higher costs, but the number of passengers using one of the trains increases as well, which can lead to extra revenue. For future research we aim to search for different approaches to determine the demand for the rolling stock circulation based on the utilizations observed in the simulation. A different approach to the $\mu + 2\sigma$ rule would be to adapt demand based on observed scores.

# 6. COMBINATORIAL ASPECTS

In the original "El-Farol Bar" model, it is not difficult to see that the ideal utilization of the bar lies at 60%, because all agents have the same payoff. In our extension it is not easy to determine the ideal utilization, as we are allowed to have agents with different scoring functions assigned to the same resource. As a result, it can be the case that for a single resource, some agents have a positive payoff and others have a negative one. The individual choice sets complicate matters even further. As a result, it is a combinatorial problem to maximize $\text{pos}(x)$. We will show this by proving the NP-completeness of the related decision problem.

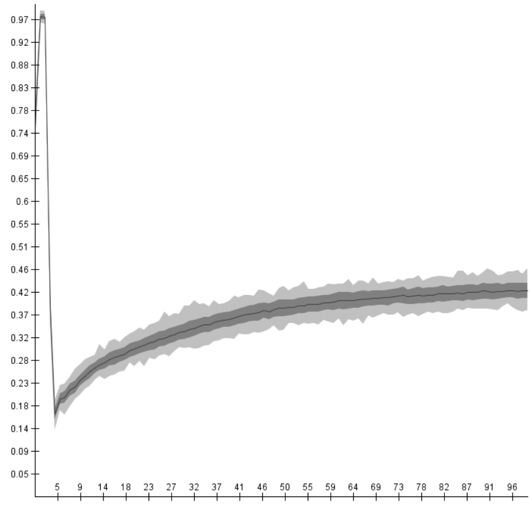
THEOREM 1. *For a given instance of the game, deciding whether there exists a valid outcome $x \in \mathcal{O}$ such that all agents have a positive payoff (i.e. whether* $\text{pos}(x)$ *is equal to* 1*) is NP-complete, even if we have threshold scoring functions with* 2 *different thresholds and we allow only singleton resources in the choice sets.*

PROOF. We will show NP-hardness by reduction from the $k$–SET COVER problem [9]. In the $k$–SET COVER problem we are given a collection $\mathcal{A} = \{A_1, \ldots, A_n\}$ of $n$ sets, a set of all elements $U = \bigcup_{i \in N} A_i$ and a positive integer $k$. We have to decide whether there exists a subset $\mathcal{A}' \subseteq \mathcal{A}$ such that $|\mathcal{A}'| \leq k$ and $\bigcup_{A \in \mathcal{A}'} A = U$.

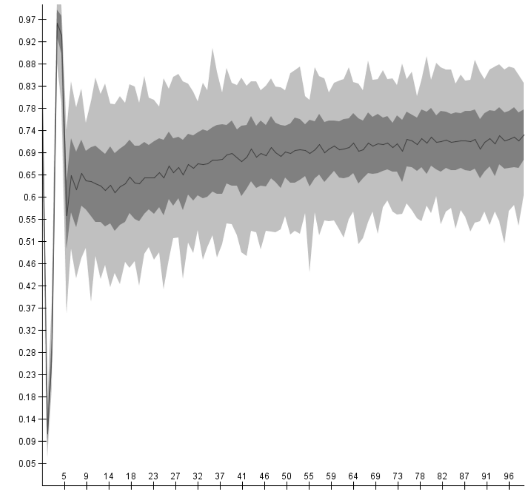
We now introduce $|U|$ regular agents and $|\mathcal{A}| - k$ grumpy agents. We introduce a mapping between the sets in $\mathcal{A}$ and the resources. Each element in $e \in U$ is represented by a regular agent which has a choice set that consists of singleton resources corresponding to the sets in $\mathcal{A}$ containing $e$. The grumpy agents have a choice set with a singleton for every resource. We define the payoff functions such that the regular agents have a positive payoff as long as they have chosen a resource, and the grumpy agents have a positive payoff if they are exclusively assigned to a resource (if we fix all $\text{cap}(r) = 1$, then $\theta_i = n$ if $i$ is a regular agent and $\theta_i = 1$ if $i$ is a grumpy agent). As a result the grumpy agents can only have a positive payoff if they are assigned to resources in such a way that all the other agents can be assigned to the remaining resources. By construction of the choice sets, this is only possible if the remaining $k$ resources that are not utilized by the grumpy agents correspond to sets that are able to cover all elements. Thus, we have reduced the $k$-SET COVER problem into our decision problem with 2 threshold scoring functions and singleton choice sets.

NP-completeness then follows from the fact that given a vector $x$, we can easily check whether it is feasible and whether indeed $\text{pos}(x) = 1$. □

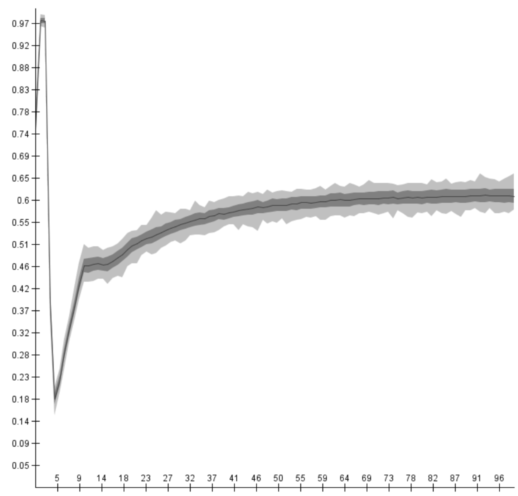
In order to understand how the reduction works, we provide an example in Table 3. Here the $A$'s and $e$'s represent the sets and elements of the $k$-SET COVER instance. The



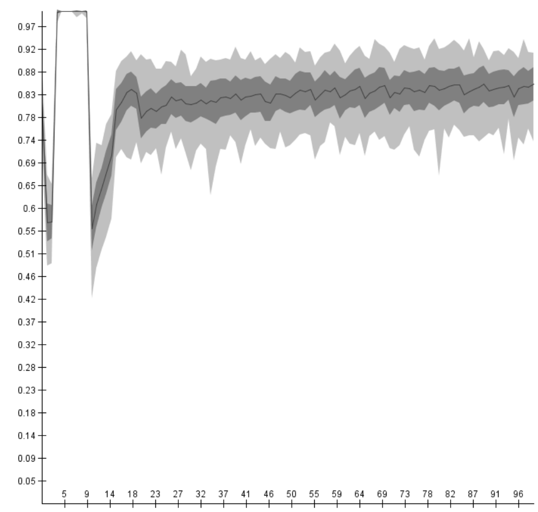
(a) utl when rescheduling period $k = 1$



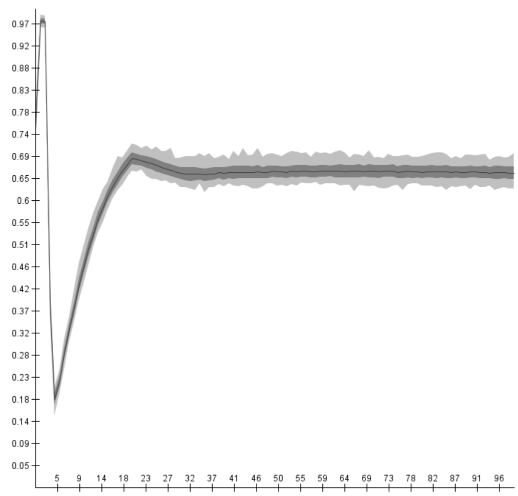
(b) posc when rescheduling period $k = 1$



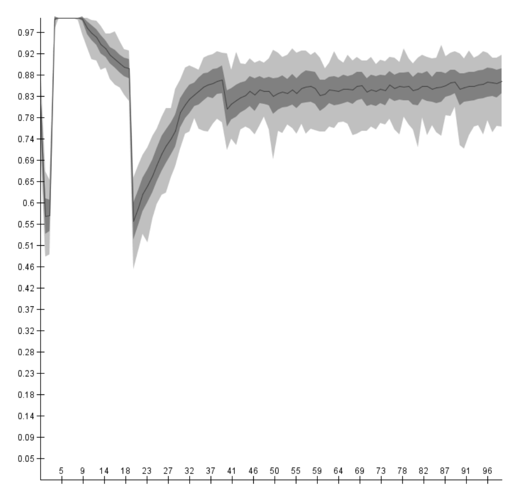
(c) utl when rescheduling period $k = 5$



(d) posc when rescheduling period $k = 5$



(e) utl when rescheduling period $k = 10$



(f) posc when rescheduling period $k = 10$

Figure 2: Results of the capacity rescheduling experiments. The dark line shows the mean over all 100 experiments, the dark gray area is one standard deviation away from the mean and the light gray area shows the minimum and maximum values observed.

Table 3: An example reduction from $k$-Set Cover to a game instance.

| | $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ |
|---|---|---|---|---|---|
| $A_1$ | × | × | | | |
| $A_2$ | | | | × | × |
| $A_3$ | | × | × | | |
| $A_4$ | | | × | × | |
| $A_5$ | × | | | | × |

(a) An example instance of a $k$-Set Cover problem.

| Agent | $C_i$ |
|---|---|
| $a_1$ | $\{\emptyset, \{1\}, \{5\}\}$ |
| $a_2$ | $\{\emptyset, \{1\}, \{3\}\}$ |
| $a_3$ | $\{\emptyset, \{3\}, \{4\}\}$ |
| $a_4$ | $\{\emptyset, \{2\}, \{4\}\}$ |
| $a_5$ | $\{\emptyset, \{2\}, \{5\}\}$ |
| $g_1$ | $\{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$ |
| $g_2$ | $\{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$ |

(b) The corresponding choice sets for $k = 3$

corresponding game instance contains the $a$ agents for the elements and $5 - k = 2$ grumpy agents denoted by the $g$ agents. For $k = 3$, we can assign the two grumpy agents to resource 4 and resource 5, as the other agents are covered by the remaining resources. If we would now change $k$ to 2, we would need to add an additional grumpy agent. However, we cannot give a positive payoff to both this additional grumpy agent and all the regular agents at the same time. This is consistent with the fact that there is no solution for the $k$-Set Cover instance if $k = 2$.

## 7. CONCLUSION AND FUTURE WORK

We have evaluated the effect of information disclosure and capacity optimization in a minority game designed to study crowding effects in public transport. The inclusion of heterogeneous agents poses many new challenges. From the theoretical perspective there are questions to what extend observations for the original minority game, such as the relation between memory length and efficiency, still apply. From the practical perspective, the question is whether an operator can influence and manage the cooperation of the agents in order to stimulate the efficient utilization of the vehicles. We have conducted two simulation studies where we focused on the practical challenges. In the first study we evaluate the effect of different information policies in a scenario where every agent uses at most a single trip every round. We find that disclosing more information attracts more agents, but that this comes at the cost of lower payoffs. This trade-off is influenced by the number of agents and the available capacity in the system.

In the second simulation study, we evaluate the effect of adaptive capacity management in the context of railway transportation. Here the agents make a journey along a line. They have to choose a time to travel between an individually assigned origin and destination stop every round. As such journeys can cross multiple stops and thus overlap on the line, more complex patterns of agent interaction can emerge. We find that the number of rounds utilizations are recorded before capacities are re-optimized has an impact on the number of agents utilizing the system and their payoffs. Rescheduling every round seems to lead to worse system performance than rescheduling every 5 or 10 rounds.

Our studies show that we are able to evaluate and compare the effects of different policies for information and capacity. The question remains whether we can improve on the policies we evaluated. We think that policies that act on agents that repeatedly have a low payoff are an interesting area for further research.

## 8. REFERENCES


[1] W. Arthur. Inductive reasoning and bounded rationality. *The American economic review*, 84(2):406–411, 1994.

[2] A. Bazzan, R. Bordini, G. Andrioti, R. Vicari, and J. Wahle. Wayward agents in a commuting scenario (personalities in the minority game). In *Proceedings of the Fourth International Conference on MultiAgent Systems*, pages 55–62. IEEE, 2000.

[3] D. Challet, M. Marsili, and R. Zecchina. Statistical mechanics of systems with heterogeneous agents: Minority games. *Physical Review Letters*, 84(8):1824, 2000.

[4] D. Challet, M. Marsili, and Y.-C. Zhang. Minority games: interacting agents in financial markets. *OUP Catalogue*, 2004.

[5] P. Fioole, L. Kroon, G. Maróti, and A. Schrijver. A rolling stock circulation model for combining and splitting of passenger trains. *European Journal of Operational Research*, 174(2):1281–1297, 2006.

[6] L. R. Ford and D. R. Fulkerson. *Flows in networks*. Princeton university press, 1962.

[7] A. Galstyan, S. Kolar, and K. Lerman. Resource allocation games with changing resource capacities. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 145–152. ACM, 2003.

[8] B. A. Huberman and R. M. Lukose. Social dilemmas and internet congestion. *Science*, 277(5325):535–537, 1997.

[9] R. M. Karp. Reducibility Among Combinatorial Problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[10] W. Kets. Learning with fixed rules: The minority game. *Journal of Economic Surveys*, 26(5):865–878, 2012.

[11] R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

[12] R. Savit, R. Manuca, and R. Riolo. Adaptive competition, market efficiency, and phase transitions. *Physical Review Letters*, 82(10):2203, 1999.

[13] R. Selten, T. Chmura, T. Pitz, S. Kube, and M. Schreckenberg. Commuters route choice behaviour. *Games and Economic Behavior*, 58(2):394–406, 2007.

# Online Routing Games and the Benefit of Online Data

László Z. Varga
ELTE Regional Knowledge Centre
Irányi Dániel u. 4.
Székesfehérvár, H-8000 Hungary
phone: +36-1-381-2186
lzvarga@inf.elte.hu

## ABSTRACT

The goal of this paper is to define and analyze the model of online routing games in order to be able to determine how we can measure and prove the benefits of online real-time data in applications like navigation by autonomous cars. Based on the models of algorithmic game theory and online mechanisms, we define the formal model of online joint resource utilization games and, as their specific version, the model of online routing games. We study simulation runs of online routing games in a Braess network, and define three different notions of the benefit of online data. We outline the possibility of different classes of online routing games and start to investigate the class of simple naïve online routing games that represents the current commercial navigation systems. We prove that in the class of simple naïve online routing games stability is not guaranteed. We prove that in simple naïve online routing games if a single flow enters the network, then the flow on some edge inside the network at some time may be bigger than the one that entered the network. As a consequence we prove that in simple naïve online routing games the worst case benefit of online data may be bigger than one, i.e. it may be a "price". By defining these frames, we open new theoretical research opportunities for important application fields.

## Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence – *Multiagent systems.*

## General Terms

Performance, Design, Economics, Experimentation, Theory.

## Keywords

online routing games, benefit of online data

## 1. INTRODUCTION

Five ongoing trends became more and more accomplished in the history of computing: ubiquity, interconnection, intelligence, delegation and human-orientation [1]. The current wave of this progress is marked by the widespread availability of online real-time data which opens new possibilities in several application areas like real-time manufacturing intelligence, industrial internet, internet of things, emergency and disaster information services, and intelligent road transport systems. The most challenging applications are those where autonomous agents have access to online real-time data and create plans how to achieve their goals in an environment where they jointly utilize resources that become more costly as more agents use them. In these applications agents are dynamically arriving and departing when they complete their plans. The plans are created by exploiting online data that describe the current status and the current cost of the resources. There is uncertainty about the feasible decision of an agent, because the cost of the resources will change by the time the agent starts to use them: departing agents will release the resources as they complete their plans, agents simultaneously creating their plans will influence each other's costs, and agents arriving later may also influence the costs of the resources used by agents already executing their plans. Because game theory is an appropriate foundation for multi-agent systems [2], we call this type of applications as *online joint resource utilization games* which we will define in this paper. Note that these games are different from resource allocation or minority games [4] which are simultaneous one shot or repeated simultaneous games where there might be some coordination among some of the agents. In contrast, online joint resource utilization games are continuous and non-cooperative games exploiting real-time online data.

A well-known online joint resource utilization game is car navigation using real-time data. In this environment traffic participants generate and use online traffic information to create their own self-optimized plan for their route which contain road sections jointly utilized with other agents. Car navigation using real-time data is a special case of online joint resource utilization games, because the allowed order of the resources in the plan of the agents is restricted by the structure of the road network. From theoretical point of view we call real-time data based car navigation applications as *online routing games*. Note that in our approach each driver makes an individual real-time data based decision when it enters the network, whereas in other approaches [5] drivers learn the best route to select, based on past selections.

Two well-known examples of real-time information based navigation systems are Google Maps and Waze. The planning in these systems is done on central server(s) which may play similar role to the virtual environment in the anticipatory vehicle routing of [3]. There are other traffic management systems that combine central planning and local freedom, like the PLANETS system [12] in which global control strategy is provided from a Traffic Management Centre, but traffic participants have a freedom to make decisions autonomously. In our view, self-interested agents will not conform to a central strategy if it is not individually rational, so the global strategy must emerge from the agents' decision. Therefore we will not have in our model an explicit concept of a central planner or a virtual environment even if the agents use the services of these abstractions.

Although it is widely believed and intuitively we might think that traffic route planning results in shorter travel time if we take into account the real-time traffic information (for example congestions), but no theoretical study is known if real-time data based car navigation produces better traffic or not. There is need for such theoretical studies, because autonomous cars are being designed and the usage of online navigation systems based on the simple naïve strategy discussed in this paper is spreading and we do not even know how to measure their benefit, not to mention how to optimize their behavior. The goal of this paper is to define and analyze the model of online routing games in order to be able to prove the possible benefits of online real-time data.

Neither online joint resource utilization games nor online routing games have been formally defined and studied in detail, so we are advancing the state of the art with the work of this paper. There is related research on routing games without online information in the algorithmic game theory field and there is research on auctions with online information in the online mechanisms field. We are combining these two fields to create the new theoretical model.

Algorithmic game theory [11] studies routing games in which end users simultaneously select a full route to their destination in a network that is susceptible to congestion. It is well known that individually self-optimizing travel routes does not necessarily result in optimal traffic (optimal for the sum of the travel times) and each participant may have longer travel time than with central planning. This is known as the "*price of anarchy*" which was explored by 2012 Gödel prize winners Roughgarden and Tardos. In their paper [9] they investigated the old conundrum in transportation science known as the "Braess's paradox" [10]. The algorithmic game theory investigations revealed important properties of routing games, however the algorithmic game theory approach includes assumptions which do not handle the dynamic online information environment.

Dynamic agent systems have been studied in the framework of online mechanisms, like in [7]. Online mechanisms [8] extend mechanism design to dynamic environments where agents continuously enter and leave the environment. Agents in online mechanisms make decisions without knowing the future. Agents have a type which is described by their time of arrival, time of departure and their valuation of the resources to be allocated. The utility of the agents is the difference between their valuation and the cost of the allocated resource. While online mechanism design is good for certain types of dynamic environments, its assumptions do not handle the decentralized nature of real-time data based car navigation systems. Here decentralization refers to both decision making by all the agents and cost value determination by the resources at the actual time of resource utilization. In online mechanisms the cost of the allocated resource is determined at negotiation time by a centralized agent, while in real-time data based car navigation the final cost is determined by the resources when the resource is actually used, and the final cost may be different from the one at decision time.

In order to be able to forecast the behavior of the dynamic agent environments of online joint resource utilization games and to be able to measure and prove the benefits of online real-time data, our contributions are the following:

Based on the models of algorithmic game theory and online mechanisms, we define the first model of online joint resource utilization games and, as their specific version, the model of online routing games.

We study the simulation of an online routing game in a Braess network and point out that there are worst, best and average benefits of online real-time data, and we define these notions. We outline the possibility of different classes of online routing games with different strategies.

We prove three properties of the class of the simple naïve strategy online routing games and the benefit of online real-time data in these games.

With these advances we open new research opportunities for important application fields, define the main characteristics that can form the basis to guide future research and help to compare the results of future research contributions.

## 2. Related Work

In this section we highlight the main theoretical findings of routing games and online mechanism design, because the first model of online joint resource utilization games presented in this paper is based on them and the research in the new field of online joint resource utilization games has to answer similar questions.

### 2.1 Routing Games

Algorithmic game theory studies networks with source routing (section 18 in [11]), in which end users simultaneously choose a full route to their destination and the traffic is routed in a congestion sensitive manner. Two models are used: nonatomic selfish routing and atomic selfish routing. Nonatomic routing is meant to model the case when there are very many actors, each controlling a very small fraction of the overall traffic. Atomic routing is meant to model the case when each actor controls a considerable amount of traffic. Both models are studied in detail and showed similar properties. The main difference is that different techniques are required for their analysis, because the nonatomic model basically has continuous functions having unique extreme values, while the atomic model has discrete functions approximating the extreme values at several points.

The algorithmic game theory model of the routing problem is the (G, r, c) triple, where

G is the road network given by a directed graph G=(V, E) with vertex set V and edge set E;

r is the total traffic flow given by a vector of $r_i$ traffic flows with $r_i$ denoting the amount of flow on the $P_i$ trip which is from the $s_i$ source vertex of G to the $t_i$ target vertex of G; and

c is the throughput characteristic of the road network given by a cost function with $c_e$: $R^+ \rightarrow R^+$ for each e edge of G mapping the total traffic on edge e to the travel time on that edge.

In this model the G graph may contain parallel edges; the $c_e$ cost functions are nonnegative, continuous and nondecreasing; the $r_i$ traffic flow on the $P_i$ trip is deterministically routed somehow on the paths leading from $s_i$ to $t_i$; the cost of a path is the sum of the costs of the edges in the path at a given flow; and the cost of an $r_i$ traffic flow on the $P_i$ trip is the sum of the cost of all the paths in $P_i$. In the case of a nonatomic routing problem the $r_i$ traffic flow on the $P_i$ trip may be divided arbitrarily among several paths leading from $s_i$ to $t_i$, while in the case of an atomic routing problem the $r_i$ traffic flow on the $P_i$ trip can be sent on one single path leading from $s_i$ to $t_i$. We assume that in the routing problem each actor is interested in an $r_i$ traffic flow on a $P_i$ trip, therefore we will use the term "actor", "agent" and "$r_i$ traffic flow" interchangeably.

A flow distribution is optimal if it minimizes the cost of the total traffic flow over all possible flow distributions. A flow distribution is an equilibrium flow distribution if none of the actors can change its traffic flow distribution among its possible paths to decrease its cost. The equilibrium flow distribution is a rational choice for every autonomic actor, because deviating from the equilibrium would increase the cost for the actor.

It is proven (section 18 in [11]) that every nonatomic routing problem has at least one equilibrium flow distribution and all equilibrium flow distributions have the same total cost. The price of anarchy is the ratio between the cost of an equilibrium flow distribution and the optimal flow distribution. If the cost functions are of the form $ax+b$, then the price of anarchy in any nonatomic routing problem is not more than $4\div3$. If the cost functions can be nonlinear, then one can create cost functions to exceed any given bound on the price of anarchy of nonatomic routing problems.

In atomic routing problems the existence of equilibrium flow distribution is not always guaranteed. Atomic routing problems have equilibrium flow distribution if every $r_i$ traffic flow has the same value or if the cost functions are of the form $ax+b$. If there are more than one equilibrium flow distributions, then their total costs may be different. If the cost functions of an atomic routing problem are of the form $ax+b$, then the price of anarchy is at most $(3+\sqrt{5})\div2$. If the cost functions of an atomic routing problem are of the form $ax+b$ and in addition every $r_i$ traffic flow has the same value, then the price of anarchy is at most $5\div2$.

It is known that if the routing problem has an equilibrium and the actors try to minimize their own cost (best-response), then the traffic flow distribution converges to an equilibrium.

The algorithmic game theory investigations of the routing game revealed important properties, however the algorithmic game theory model contains the following assumptions:

a) the throughput characteristic of the network does not change with time and the drivers can compute this characteristic or learn it by repeatedly passing the road network;

b) the drivers simultaneously decide their optimal route; and

c) the outcome travel time for a given driver depends on the choice of all the drivers and the characteristic of the network, but not on the schedule of the trip of the drivers.

These assumptions do not completely describe car traffic where the drivers use car navigation with online data, in which case

a) the throughput characteristic of the network changes with time and the drivers cannot compute or learn this characteristic by repeatedly passing the road network, because there may be an accident on a road and the road becomes susceptible to congestion, and then later when the accident is cleared the road is less susceptible to congestion;

b) the drivers do not decide their route at the same time simultaneously, because drivers continuously enter the road network and decide their optimal route when they enter the road network and the decision is based on the current live information about the status of the road network; and

c) the outcome travel time for a given driver depends not only on the current characteristic of the network and the route choice of all the drivers currently entering the road network, but also on the trip schedule of other drivers that entered the network previously, are currently entering the network or will enter the network later.

The issue of traffic dynamism is studied in the field of dynamic traffic assignment [6], but there they investigate the time-varying properties of traffic flow, whereas here we assume that the traffic flow is constant and only the cost functions may change. In our investigations the critical issue is the sequential decision making of the agents. This partly handled by online mechanisms.

## 2.2 Online Mechanisms

Online mechanism design problem is a multi-agent sequential decision making problem. When agents participate in the mechanism, they report to a central planner for a given period their request for certain resources at given valuations (which may be different from their private values). The central planner decides which resources at which cost are allocated to which agent in each time step. All agents are trying to maximize their utility.

The model of the online mechanism problem is the $(t, \Theta, k, c, u)$ five tuple, where

$t=\{1, 2, \dots\}$ is a possibly infinite sequence of time periods;

$\Theta$ is the set of agent types where each agent type is characterized by the $\theta_i =(a_i,d_i,v_i(t))$ triple where $a_i\in t$ is the arrival time of the agent, $d_i\in t$ is the departure time of the agent, and $v_i(t)$ is the valuation function of the agent in time period $t\in[a_i,d_i]$, the agent has no value for $t\notin[a_i,d_i]$;

$k=(k^1, k^2, \dots )$ is a sequence of decision vectors with $k^t=(k^t_1, k^t_2, \dots)$ decision vector made in time period $t$ and $k^t_i$ the decision made for agent $a_i$;

$c$ is the cost function of the decisions and $c(k^t_i)$ is the cost for agent $a_i$ in time period $t$;

$u$ is the utility function of the agents where $u_i(t)=c(k^t_i) - v_i(t)$ is the utility for agent $a_i$ in time period $t$, and all agent aim to maximize their utilities;

In this model the $\Theta$ set of agent types may be model free when no probabilistic information is known about the agents, or may be model based if probabilistic information is known. The agents may report values different from their private agent type, but only for the time period when they are present, at the beginning of the reported time period and without knowing the reports of the other agents (closed direct revelation). Usually the goal is to design online mechanisms where the truthful revelation is the dominant strategy. The effectiveness of online mechanisms is measured similarly as that of online algorithms: the performance of the online mechanism is compared with that of an offline mechanism that has the complete information about all future agent types.

The dynamic nature of online mechanisms is a good starting point to model online joint resource utilization games, however the differences are considerable: in contrast with online mechanisms, there is no central planner (agents make their own plan), there is no arrival and departure time (agents want to start their plan when they arrive), and the actual cost is determined not at the decision time, but at utilization time.

# 3. The Model of Online Joint Resource Utilization Games

In order to have a generic model, we are now defining the model of the online joint resource utilization game as an extension of the algorithmic game theory model of the routing problem and the online mechanisms. The model resembles the algorithmic game theory routing game model in the concepts of flow, cost and resource, and it resembles the model of online mechanisms in the sequences of time periods and decisions. T time unit is introduced in order to be able to compute the rate of resource utilization.

The model of the online joint resource utilization game is the (t, T, G, c, r, k) sextuple, where

t={1, 2, …} is a possibly infinite sequence of time periods;

T is a natural number, T time periods give one time unit;

G is a set of resources where each $e \in G$ resource is characterized by a $c_e$ cost function and $t_e$ resource utilization time;

c is the cost function of the resources with $c_e: R^+ \to R^+$ for each $e \in G$ mapping the resource utilization flow (the total number of agents starting to use the e resource from time period t-T to time period t) to the current cost of the resource;

r is the total resource utilization flow given by a vector of $r_i$ resource utilization flows with $r_i$ denoting the resource utilization flow (number of agents in T time periods) for the $P_i$ plan which is a set of vectors containing elements from G;

$k=(k^1, k^2, \dots)$ is a sequence of decision vectors with $k^t=(k^t_1, k^t_2, \dots)$ decision vector made in time period t and $k^t_i$ the decision made by the agents of the $r_i$ resource utilization flow in time period t.

In this model the $c_e$ cost functions are nonnegative, continuous and nondecreasing; the $c_e$ cost functions have a fixed minimum value plus a flow dependent part, where the flow dependent part is not known to any of the agents of the model and the agents can learn the actual cost only when an agent finishes using the resource and reports its cost; the $r_i$ resource utilization flow is given by $T \div n_i$ where $n_i$ is a natural number constant, meaning that one agent enters the game in each cycle of $n_i$ time periods (0 agent entering the game in time periods 1, 2, … $n_i$-1 and 1 agent entering the game with the goal of $P_i$ plan in the $n_i$ time period); the $k^t_i$ decision is to instantiate the $P_i$ plan to one of its vectors; and the actual cost of a plan instantiation ($e_1$, $e_2$, $e_3$, …) for a flow starting at time period t is $c_{e1}(t)+c_{e2}(t+t_{e1})+c_{e3}(t+t_{e1}+t_{e2})+\dots$ , i.e. the actual cost of a resource is determined at the time when the usage of that resource starts.

## 3.1 The Model of Online Routing Games

We are now defining the model of online routing games as online joint resource utilization games with a restriction on the allowed plans represented by a graph and with somewhat different cost functions. The typical application of online routing games is real-time data based car navigation where the graph represents a road network, the agents represent the cars, resource utilization means passing a road section and the cost of resource utilization represents the travel time on a road section.

The model of the online routing game is the (t, T, G, c, r, k) sextuple, where

t, T and k are the same as in online joint resource utilization games;

G is a directed graph G=(V, E) with vertex set V and edge set E where each $e \in E$ is characterized by a $c_e$ cost function which is equal to its utilization time;

c is the cost function of G with $c_e: R^+ \to R^+$ for each e edge of G mapping the flow to the travel time on that edge, but not less than the remaining cost of any other agents currently utilizing that edge increased with the time gap of the flow of the agent[1];

r is the total flow given by a vector of $r_i$ flows with $r_i$ denoting the flow aiming for a $P_i$ trip from a $s_i$ source vertex of G to a $t_i$ target vertex of G;

In this model the G graph may contain parallel edges. The $c_e$ cost functions are nonnegative, continuous and nondecreasing, their variable part are not known to any of the agents of the model and the agents can learn the actual cost only when an agent exits an edge and reports it; the $r_i$ flow is given by $T \div n_i$ where $n_i$ is a natural number constant. The $k^t_i$ decision is how the $P_i$ trip is routed on a single path of the paths leading from $s_i$ to $t_i$ and the actual cost of a path ($e_1$, $e_2$, $e_3$, …) for a flow starting at time period t is $c_{e1}(t)+c_{e2}(t+c_{e1}(t))+c_{e3}(t+c_{e1}(t)+c_{e2}(t+c_{e1}(t)))+\dots$ , i.e. the actual cost of an edge is determined at the time when the flow enters the edge.

The agents can learn the actual cost of the edges only when an agent finishes using the resource and reports its cost. Because agents do not report cost values in each time step, the agents interested in the cost values must do reasoning and decrease the last reported value by taking into account the time elapsed since the last reporting event (it is like pheromone evaporation in [16]).

The online routing game model can accommodate changes of the c cost function over the t sequence of time periods, because the agents can get information about the actual cost only from the cost reported by the agents exiting an edge.

### *3.1.1 Routing strategy*

The critical point in the online routing game is how to determine the best decision vector k. The algorithmic game theory approach assumes that the agents have full information about the cost functions and the theory tells what the best strategy is in the case of simultaneous decisions, but does not tell how the agents can achieve this. In online mechanisms a central planner decides which resources at which cost are allocated to which agent. In online routing games there is no central planner. The agents in online routing games will have to apply algorithms similar to online algorithms [13]. At this time we are not investigating how the agents of online routing games determine their strategy, instead we are investigating how current navigation systems perform in online routing games.

In practice, typical navigation software in cars use simple shortest path search in the road network, possibly modifying the distances with the online information about the actual traffic delay. We call this decision strategy as *simple naïve strategy*. We are investigating this strategy in this paper because of its practical importance. Note that the simple naïve strategy is by definition deterministic, thus it is a pure strategy.

Although online routing game has some resemblance to a sequence of atomic routing problems of the algorithmic game theory approach, online routing game is much more complex than the atomic routing problem and we do not know any theoretical results regarding the existence of equilibrium flow or something like the price of anarchy in online routing games.

[1] In this model cars cannot overtake the cars already on the road and there is a time gap, i.e. minimum "following distance".

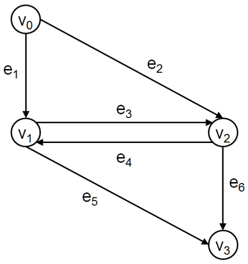



**Figure 1. The $SN_{Braess}$ network.**

## 4. Simulation of an Online Routing Game

Several authors investigated with simulation tools how the traffic would behave if the majority of vehicles used traffic information in their route planning and concluded that online data has to be used carefully in traffic scenarios [15]. These investigations did not have theoretical conclusions. In order to have a better understanding of online routing games and formally prove what others suspected from empirical investigations, we are demonstrating simulation runs in a small instance of online routing games which we call Simple Naïve Braess ($SN_{Braess}$) online routing game, because it investigates the real-time data based car navigation problem on a network corresponding to the Braess paradox [10] and the decision mechanism uses the simple naïve strategy. We selected the $SN_{Braess}$ online routing game, because the simple naïve strategy has practical importance and because the Braess paradox is a distinguished study area of transportation science and algorithmic game theory. The analysis of simulation runs in this paper is not a full statistical analysis, because the goal is to understand the behavior of the agents.

The $SN_{Braess}$=(t, T, G, r, c, k) instance of online routing games used in the simulations has the following concrete values

T=600, modeling one minute.

G is the road network (shown in Figure 1. ) is given as a four node directed graph with V=($v_0$, $v_1$, $v_2$, $v_3$) and E=($e_1$, $e_2$, $e_3$, $e_4$, $e_5$, $e_6$). The edges are $e_1=v_0\rightarrow v_1$, $e_2=v_0\rightarrow v_2$, $e_3=v_1\rightarrow v_2$, $e_4=v_2\rightarrow v_1$, $e_5=v_1\rightarrow v_3$, and $e_6=v_2\rightarrow v_3$. Note that the $e_4$ edge is included to allow bidirectional travel between $v_1$ and $v_2$ and to have an uncongested route from the source to the destination.

r=($r_1$) is the total traffic flow with only one flow on the $P_1$ trip from the $v_0$ source vertex of G to the $v_3$ target vertex of G;

c is the cost function of the road network with $c_{e1}=1+x\div10$, $c_{e2}=15$, $c_{e3}=7.5$, $c_{e4}=7.5$, $c_{e5}=15$, $c_{e6}=1+x\div10$, where x is the total number of agents entering an edge from time period t-T to time period t. As discussed in the previous section, the variable part of the cost function is not known to the agents of the model.

k=($k^1$, $k^2$, … ) is a sequence of decision vectors with $k^t=(k^t{}_1)$ decision vector made in period t. The decision is a simple naïve decision mechanism which is based on the currently reported costs and selects the path on the $P_1$ trip that currently has the minimum reported cost. The $P_1$ trip has the {$p_1$, $p_2$, $p_3$, $p_4$} set of paths, where $p_1=(e_1,e_5)$, $p_2=(e_1,e_3,e_6)$, $p_3=(e_2,e_6)$, $p_4=(e_2,e_4,e_5)$ and the costs of the paths used for decision making are the sum of the cost of the edges at the time of decision making, e.g. $c_{p1}(t)=c_{e1}(t)+c_{e5}(t)$.

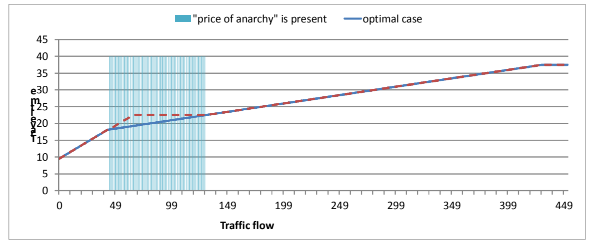



**Figure 3. Cost in the $SN_{Braess}$ network depending on the flow using the algorithmic game theory approach.**

From algorithmic game theory point of view the Braess road network (without online data) is used to demonstrate the "paradox" that the equilibrium flow has a price of anarchy with at most $4\div3$ ratio. The "paradox" is that if nonatomic selfish flow allocation is used, then at some flow values the $e_3$ edge increases the total cost ("price of anarchy"), and without the $e_3$ edge (seemingly smaller throughput) the cost is smaller (faster travel). The cost values (in this case travel times) of the trips in the $SN_{Braess}$ network from the algorithmic game theory point of view are shown with red dashed line in the diagram of Figure 3. as a function of the incoming traffic flow. The figure shows the optimal case of central planning (blue line) as well. When there is a difference, then "price of anarchy" is present.

We have investigated the cost values in the $SN_{Braess}$ online routing game by simulation runs. A constant traffic flow was entered into the network from time zero and the history of the travel time of the cars arriving at their destination at node $v_3$ during a simulation run of 411000 time periods were recorded in a diagram like in Figure 2. These travel time diagrams revealed different patterns depending on the amount of traffic flowing into the network. In some cases they had higher frequency variation with some sudden peaks, while in other cases the variations seemed to be smoother with some disturbance periods. Figure 2. shows the travel time diagram at incoming traffic flow of 120, which represents a heavy load in the "price of anarchy" range in the algorithmic game theory approach. This diagram, like the others, has a starting phase where the travel time is increasing as more and more cars are on the road. Later, as the shortest path gets congested, the drivers deviate and keep deviating to other routes, so the travel time does not seem to converge to a constant value. The travel time varies between 16 and 46.5 with an average of 30.73.

Note that in this $SN_{Braess}$ road network a) if the routes are planned with optimal central control, then the travel time is 22.0, b) if the routes are planned in accordance with the algorithmic game theory approach, then the travel time is 22.5 (with price of

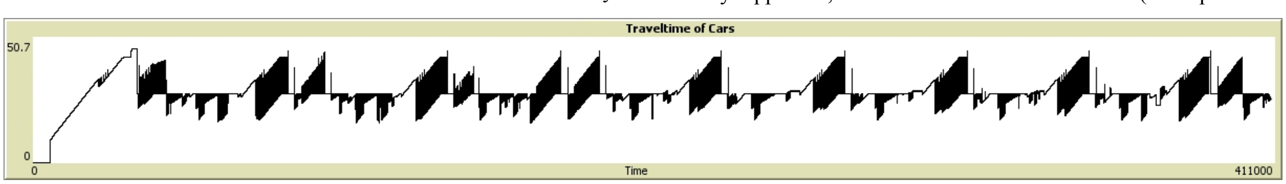



**Figure 2. The diagram of the travel time of the cars in the $SN_{Braess}$ road network at incoming traffic flow 120.**

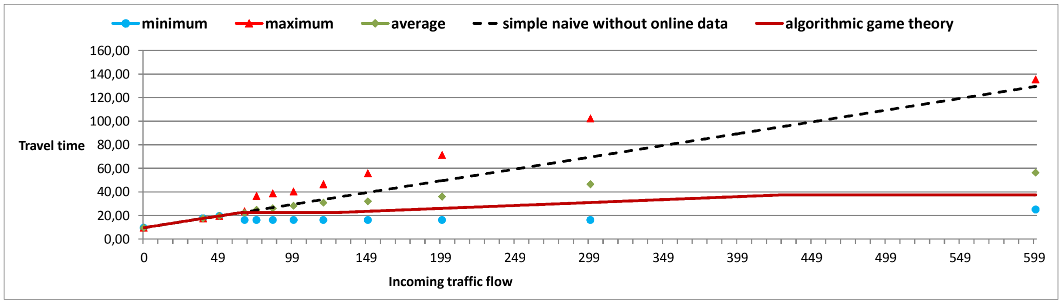



**Figure 4. Maximum, minimum and average travel times in the $SN_{Braess}$ road network at selected incoming traffic flows.**

anarchy), c) if the routes are planned with simple naïve strategy without real-time congestion delay information (i.e. all cars select the $p_2$ path), then the travel time is 33.5, and d) if all cars select the $p_4$ path which is not susceptible to congestion and is the longest path without congestion delay, then the travel time is 37.5.

We cannot include too many travel time diagrams in this paper, instead we sum up the measured maximum, minimum and average travel time values at some selected incoming flows in graphical form on Figure 4. This figure includes the travel time for the algorithmic game theory approach (continuous red line) and the travel time for the simple naïve strategy that does not use online data (dashed black line).

We can observe that at low incoming traffic flows the cars do not deviate from the shortest path, so the computed and the $SN_{Braess}$ travel times are the same. At higher flows there is fluctuation and the cars sometimes experience fast traffic, but often they experience considerable congestion delay and travel longer than the travel time of the simple naïve strategy without real-time congestion delay information.

## 5. The Benefit of Online Real-time Data

We would like to be able to tell if the agents are better off by making decisions based on online real-time data or not. In order to be able to compare the costs of the agents using online data with the costs of the agents not using online data, we have to know what we are going to compare with what.

*Oracle based evaluation*

If we take the approach of online algorithms, then we would compare the results of the online routing game with the results of an oracle that has all the information needed. One might think that in our case the oracle with all information would be the central planning[2], because the central planner has all the information and can tell each agent which route to take. The central planner produces the travel times as shown by the blue line in Figure 3.

The central planning oracle might be good to measure the global effectiveness of the agents in the $SN_{Braess}$ model, however it evaluates not only the benefits of making decisions based on online data, but in addition it evaluates the different decision making strategies as well. In the $SN_{Braess}$ model there is no coordination among the agents and the agents make online decisions using the simple naïve strategy, while in the central planning and the algorithmic game theory approach the agents are coordinated and exploit their knowledge about the cost functions. Therefore if we want to evaluate only the benefits of online real-time data, then we want to compare the results with an "oracle" using the same decision making strategy.

*Oracle with the same decision making strategy*

The decision making strategy of the $SN_{Braess}$ model is to select the path with the smallest cost (shortest travel time) using the current online real-time data and the agents do not have a cost model of the network. The corresponding decision strategy without online real-time data is the one which uses the shortest path without the variable part of the cost functions. The dashed black line in Figure 4. shows this oracle. As we can see in Figure 4. the agents of the $SN_{Braess}$ model perform much better in some cases (minimum values), in average (average values) they perform better at higher incoming traffic flow range and sometimes in the worst case (maximum values) they perform worse, mainly in the medium flow range.

*Best, worst and average*

In the algorithmic game theory model there is equilibrium and the price of anarchy concept is the ratio between the equilibrium and the optimum. In the simulation runs of the $SB_{Braess}$ model there are maximum, minimum and average cost values at every incoming traffic flow. Later in this paper, in Theorem 1 we will prove that there are simple naïve strategy online routing games which do not have equilibrium at some flow values. If there is no equilibrium, then we must have different measures of the benefit of online real-time data for the best, worst and average cases (which are guaranteed to exist if there are finite sequence of time periods).

Depending on the type of application, we are interested in the different types of benefits. The most important is the worst case, because it can be used to provide a guarantee in critical applications. The best case can be used in applications, where we have to make sure that a certain value is achieved at least once. The average case is seldom useful in itself, usually we have to consider other statistical distribution parameters as well.

*Measure of the benefit of online real-time data*

We conclude the above discussion with defining the different benefits of online real-time data. If these benefits are greater than 1, then they are in fact a "price" like the price of anarchy.

*Definition 1.* The *worst case benefit* of online real time data at a given flow is the ratio between the cost of the maximum cost of the flow and the cost of the same flow with an oracle using the

[2] It is assumed that the central planner is able to produce the optimal flow distribution as defined on page 463 in [11].

same decision making strategy and only the fixed part of the cost functions.

*Definition 2.* Similarly, the *best case benefit* of online real time data at a given flow is the ratio between the cost of the minimum cost of the flow (after the initial running up) and the cost of the same flow with an oracle using the same decision making strategy and only the fixed part of the cost functions.

*Definition 3.* The *average case benefit* of online real time data at a given flow is the ratio between the cost of the average cost of the flow and the cost of the same flow with an oracle using the same decision making strategy and only the fixed part of the cost functions.

## 6. Classes of Online Routing Games

In the above sections we have seen that the decision strategy is important in online routing games. The aim is to select a decision strategy that results in costs close to the optimum. Although the above discussed simple naïve decision strategy is often applied in real world, it is not the best, because it does not alternate the agents of a flow among two or more paths, whereas the optimal central planning and the algorithmic game theory approach use several paths for the same flow.

Further research is needed to study different online routing game decision strategies derived from other related games. In this paper we are only mentioning a possibility. In addition to the already discussed shortest path planning, we can start the development of decision strategies from resource allocation or minority games [4]. These games are simultaneous games containing some coordination as well, but in online routing games coordination is excluded. The coordination aspect could be replaced by some probabilistic value and each agent would randomly select its path in the network based on a probabilistic distribution. The optimal probabilistic distribution values could be determined using a kind of evolutionary algorithm like the one applied to the El Farol Bar problem in [14].

Online routing games using the same type of decision strategies belong to the same class of online routing games. Each class need to be evaluated how much benefit they make out of online real-time data, in order to be able to determine the type of application where they are suitable. The evaluation should include formal proofs. In this paper we are formally analyzing the class of simple naïve strategy online routing games.

## 7. Three Properties of the Class of Simple Naïve Strategy Online Routing Games

We are now proving three properties of the simple naïve strategy online routing games.

THEOREM 1. *There are simple naïve strategy online routing games which do not have equilibrium at certain flow values.*

PROOF. We show some games that satisfy this claim. Let $SN_{7.1}$ be a (t, T, G, r, c, k) simple naïve strategy online routing game. Let $r=(r_1)$ be the total traffic flow with only one flow on the $P_1$ trip of G. Let $P_1=\{p_1,p_2\}$ with $p_1$ and $p_2$ two different paths of $P_1$. Let $c_{nc1}$ be the cost of $p_1$ when there is no congestion on $p_1$ and $c_{c1}$ be the cost of $p_1$ when $r_1$ flows on $p_1$. Similarly let $c_{nc2}$ be the cost of $p_2$ when there is no congestion on $p_2$ and $c_{c2}$ be the cost of $p_2$ when $r_1$ flows on $p_2$. There might be cost functions such that $c_{c1}>c_{c2}>c_{nc1}>c_{nc2}$ at $r_1$ flow. In this case there is no equilibrium, because in the beginning $p_2$ is selected by all agents, but as soon as the cost of $p_2$ goes above $c_{nc1}$, all agents select $p_1$, so $p_2$ becomes less congested and, as a result, the cost of $p_2$ will drop below the cost of $p_1$, so all agents will select $p_2$ again, and the cycle starts again. □

THEOREM 2. *There are SN=(t, T, G, r, c, k) simple naïve strategy online routing games where the total traffic flow has only one incoming flow, i.e. $r=(r_1)$, however the flow on some of the edges of G sometimes may be more than $r_1$.*

PROOF. We show some games where this is possible. Let $SN_{7.2}$=(t, T, G, r, c, k) be a simple naïve strategy online routing game. Let $r=(r_1)$ be the total traffic flow with only one flow on the $P_1$ trip of G. Let $P_1=\{p_1,p_2\}$ with $p_1$ and $p_2$ two different paths of $P_1$. Let $p_1=(e_1,e_3)$, $p_2=(e_2,e_3)$ be the edges of the paths. Let $e_2$ be an edge not susceptible to congestion, $c_2$ be the cost of $e_2$, and $c_2>2\times T$. Let $e_3$ edge be susceptible to congestion. Let the $e_1$ edge be susceptible to congestion and let the cost of $e_1$ be such that $c_{nc1}$ is the cost of $e_1$ when there is no congestion on $e_1$ and $c_{c1}$ is the cost of $e_1$ when $r_1$ flows on $e_1$, and $1.5\times c_2>c_{c1}>c_2>c_{nc1}$. This is possible at some $r_1$ flow on the edge for example if the cost function is linear. When $r_1$ starts to flow into G, then it goes on the $p_1=(e_1,e_3)$ path, because $c_2>c_{nc1}$. In the beginning the cost of $e_1$ is $c_{nc1}$, but it is increasing and at some $t_{xc1}$ time (where $t_{xc1}<T$) the cost of $e_1$ reaches $c_2$. The $c_2$ travel time will be reported to other agents when these agents exit $e_1$, i.e. at $t_{xc1}+c_2$ time. From this time agents will select the $p_2=(e_2,e_3)$ path. This flow from $e_2$ will reach the $e_3$ edge at $t_{xc1}+c_2+c_2$ time. We are going to show that at this time the flow from $e_1$ to $e_3$ already reached the $r_1$ value and it is still $r_1$. At time T the agents are still selecting the $e_1$ edge, because $T<0.5\times c_2$ and $0<t_{xc1}$, so $T< t_{xc1}+c_2$, the path change time. Therefore the flow from $e_1$ to $e_3$ does reach the $r_1$ value at $T+c_{c1}$ time. Because $T<0.5\times c_2$, $c_{c1}<1.5\times c_2$ and $0<t_{xc1}$, we get $T+c_{c1}<t_{xc1}+c_2+c_2$, so the flow from $e_1$ to $e_3$ already reached the $r_1$ value when agents start to flow from $e_2$ to $e_3$. The last agent selects the $e_1$ edge at $t_{xc1}+c_2$ time and because $0<t_{xc1}$ and $T<0.5\times c_2$, this agent starts after T at full $r_1$ flow therefore the cost for this agent is $c_{c1}$. Therefore the flow from $e_1$ to $e_3$ does not stop until $t_{xc1}+c_2+c_{c1}$. Because $c_2<c_{c1}$, we get $t_{xc1}+c_2+c_2<t_{xc1}+c_2+c_{c1}$, so the flow from $e_1$ to $e_3$ still has the $r_1$ value when agents start to flow from $e_2$ to $e_3$. So the $e_3$ edge will receive $r_1$ flow from the $e_1$ edge and some additional flow from the $e_2$ edge at $t_{xc1}+c_2+c_2$ time, and the cost of the agents coming from the $e_1$ edge is $c_{c1}$. (Note that it is enough if G contains a sub-graph as described in the proof of Theorem 2 and the $r_1$ flow continuously flows into this sub-graph.) □

THEOREM 3. *There are SN=(t, T, G, r, c, k) simple naïve strategy online routing games where the worst case benefit of online real-time data is greater than one, i.e. in these games the worst case benefit is a "price".*

PROOF. Let us take the Let $SN_{7.2}$=(t, T, G, r, c, k) simple naïve strategy online routing game example from Theorem 2 and let the cost of $e_3$ be such that $c_{nc3}$ is the cost of $e_3$ when there is no congestion on $e_3$, $c_{c3}$ is the cost of $e_3$ when $r_1$ flows on $e_3$. We know from Theorem 2 that there is some time when agents with $c_{c1}$ cost from the $e_1$ edge will flow into the $e_3$ edge with $r_1$ flow, and at the same time another flow will flow into the $e_3$ edge, so the cost of these agents on the $e_3$ edge will be some $c_{c3+}$, which is greater than $c_{c3}$, because the cost functions are non-decreasing. So the total cost of the agents on the $p_1$ path will be $c(p_1)=c_{c1}+c_{c3+}$. The oracle with simple naïve strategy without online data would send all the time all traffic on $p_1$ with $c(p_1)=c_{c1}+c_{c3}$ cost, so the worst case benefit of online real-time data is $(c_{c1}+c_{c3+})\div(c_{c1}+c_{c3})$. Because $c_{c3+}>c_{c3}$, the worst case benefit of online real-time data is greater than one at this flow. □

## 8. CONCLUSION

Information and communication technologies allow that modern car navigation systems utilize live online data from traffic networks to optimize the route of autonomic vehicles. Several authors investigated with simulation tools how the traffic would behave if the majority of autonomic vehicles based their route planning on such navigation systems and concluded that online data has to be used carefully in traffic scenarios. In order to be able to measure and prove properties of traffic routing based on online data, we have defined the formal model of online joint resource utilization games and, as their specific version, the model of online routing games. To our knowledge, we are the first to define these models. These models are extensions of the models of routing games of the algorithmic game theory approach and the online mechanisms. Based on the formal model, we analyzed the simulation of a simple routing scenario in a Braess network and pointed out the different aspects of the benefit of online data, and defined three notions of the benefit of online data. We foresee different classes of online routing games, among them the class of simple naïve online routing games currently applied in commercial products. We proved that in the class of simple naïve online routing games stability is not guaranteed, so it makes sense to talk about worst, average and best benefit of online data. We proved that in simple naïve online routing games it may happen that a single flow enters the network and on some edge inside the network the flow is bigger than the one that entered the network. As a consequence we proved that the worst case benefit of online data may be bigger than one, i.e. it may be a "price". These results are in line with previous simulation results, but now we have given formal proofs.

With these advances we opened new research opportunities for important application fields and determined the main notions and characteristics that can become the basis to guide future research. We challenge future research to develop online routing game decision strategies that have worst case benefits of online data below one, or prove that it is not possible to develop such strategies. If such strategies are possible, then we expect that the application of these new strategies will be individually rational choice and therefore the decision strategies can be implemented in the navigation devices themselves instead of the centralized planning approaches like those of Google Maps and Waze, because some users are reluctant to provide private data for the centralized approach.

## 9. ACKNOWLEDGMENTS

This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013).

## 10. REFERENCES


[1] Michael Wooldridge, *An Introduction to MultiAgent Systems (2nd ed.)*, Wiley Publishing, 2009

[2] Jeffrey S. Rosenschein, "Multiagent Systems, and the Search for Appropriate Foundations" in *Proc. of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), Saint Paul, Minnesota, USA, pp. 5-6., 2013

[3] R. Claes, T. Holvoet, and D. Weyns, "A decentralized approach for anticipatory vehicle routing using delegate multi-agent systems" in *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 364-373, 2011.

[4] Galstyan, A.; Kolar, S.; and Lerman, K., "Resource Allocation Games with Changing Resource Capacities" in *Proc. of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2003)*, Melbourne, Australia, page 145--152, 2003

[5] Bazzan, A. L. C.; Klügl, F. "Case studies on the Braess Paradox: Simulating route recommendation and learning in abstract and microscopic models" in *Transportation research. Part C, Emerging technologies*, vol. 13, no. 4, pp. 299-319, 2005

[6] "Dynamic Traffic Assignment: A Primer", in *Transportation Research Circular Number E-C153*, Transportation Research Board, 500 Fifth Street, NW, Washington, DC 20001, 2011

[7] E.H. Gerding, V. Robu, S. Stein, D.C. Parkes, A. Rogers and N.R. Jennings, "Online Mechanism Design for Electric Vehicle Charging" in *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), Taipei, Taiwan, pp. 811-818., 2011

[8] D.C. Parkes. "Online mechanisms" in [11] pages 411–439

[9] Tim Roughgarden and Éva Tardos, "How bad is selfish routing?" *J. ACM* 49, 2 (March 2002), pp. 236-259

[10] Braess, D. "Über ein Paradoxon aus der Verkehrsplanung," *Unternehmensforschung* 12 (1968), 258.

[11] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani, *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA. 2007

[12] J. Görmer, J. F. Ehmke, M. Fiosins, D. Schmidt, H. Schumacher, and H. Tchouankem, "Decision support for dynamic city traffic management using vehicular communication" in *Proc. of 1st International Conference on Simulation and Modeling Methodologies, Technologies and Applications*. SciTePress Digital Library, pp. 327–332, 2011

[13] S. Albers, "Online algorithms," in *Interactive Computation: The New Paradigm* edited by D.Q. Goldin, S.A. Smolka and P. Wegner, Springer, pp. 143-164, 2006.

[14] Galafassi, Cristiano and Bazzan, Ana L. C., "Evolving Mechanisms in Boolean Games", in *Proc. of Multiagent System Technologies, 11th German Conference, MATES 2013*, Koblenz, Germany, pp 73-86., 2013

[15] Joachim Wahle, Ana Lúcia C Bazzan, Franziska Klügl, Michael Schreckenberg, "Decision dynamics in a traffic scenario" *Physica A: Statistical Mechanics and its Applications*, Volume 287, Issues 3–4, Pages 669–681, 2000

[16] Dallmeyer, J., Schumann, R., Lattner, A. D. and Timm, I. J., "Don't Go with the Ant Flow: Ant-inspired Traffic Routing in Urban Environments", in *7th International Workshop on Agents in Traffic and Transportation* (ATT 2012) held at AAMAS 2012, pp. 59--68. 2012

# Learning-based traffic assignment: how heterogeneity in route choices pays off

Ana L. C. Bazzan
Instituto de Informática – UFRGS
Caixa Postal 15064, 91.501-970 Porto Alegre, RS, Brazil
bazzan@inf.ufrgs.br

## ABSTRACT

An important stage in traffic modeling and planning is traffic assignment. For this, mainly an aggregate perspective has been taken, in which zonal data is considered. In contrast, if individuals are considered as active and autonomous agents, instead of having a central component assigning trips to links, agents do their actual route choices. This disaggregate perspective yields choices that are more heterogeneous because there is no batch assignment. A consequence is that the agents are able to distribute themselves in the network, thus using it in a better way. In this paper, a disaggregate, agent-based perspective is taken in which agents learn to select routes by selecting links at each node of the network, thus also addressing en-route changes in the known routes. To illustrate this approach, a non-trivial network is used and the results are compared to iterative methods that approximate the user equilibrium.

## 1. INTRODUCTION

Traffic assignment is an important stage in the task of modeling and simulating a transportation system. It connects the physical infrastructure and the demand that is going to use it, i.e., it assign trips to each link of the road network. Thus it appears as one of the stages in the so-called "four-stage models" of traffic modeling. Specifically, it is the fourth stage, the previous three being: trip generation, trip distribution, and modal split. The present paper deals with that last stage, hence how trips are generated (a function of attractiveness of certain zones of the network), their distribution (how many trips per zone), and modal split are not addressed (in fact, this paper only deals with vehicular traffic so that other modes are not relevant).

Classical methods for traffic modeling – including trip assignment – normally adopt an aggregate perspective, i.e., zone-based instead of individual-based. The reason is that it is simpler to get zonal data (how many trips originate or terminate there) than individual data, which may also include which intermediate activities each road user does during the trip, which knowledge it has, as well as its preferences for routes. Aggregate modeling assumes a centralized entity that controls those four stages. Hence, trips are *generated*, *distributed*, *split*, and *assigned*. In contrast, in a disaggregate perspective, one talks about trip *choice*, destination *choice*, mode *choice*, and route *choice*, in opposition to generation, distribution, split, and assignment respectively.

The disaggregate perspective naturally fits an agent-based approach and it is the one followed here. In it, agents do the actual choice (instead of being told which trip to make, which destination to go, etc.). Specifically, for the assignment stage, this means that each agent will choose its route based on *local*, partial knowledge. This may look trivial but makes a difference in terms of which knowledge must be available when one decides for an aggregate versus disaggregate approach. Moreover, it also means that the choices are as heterogeneous as possible. Ultimately, each agent can decide which route to take based on its individual behavioral rule. As this is a very complex approach (it is questionable if such behaviors can be collected at all, at least with the kind of technology and sensors that we have at this stage), the perspective in the present paper is that agents are heterogeneous only regarding the information they have, but not yet regarding completely heterogeneous behavioral rules. Of course, an intermediate situation could be that classes of agents with different behavioral rules could be modeled, as in discrete choice modeling [5]. However, since even this kind of data is not always available (e.g., how many percent of the agents are greedy, etc.), this paper assumes an homogeneous population w.r.t. behavior. Also, having classes of agent in the model would mean that this should be validated against some real-world situation for which the data is not available.

As mentioned, this paper takes an agent-based, disaggregate perspective for trip assignment. The selection of route is made by each agent, based on a reinforcement learning (RL) method. This means that agents have knowledge about the travel time for the shortest path between their origins and destinations, given an uncongested state of the network. However, they may explore other alternatives as well. Given that congestion may arise, this exploration is likely to make they exploit other routes. With this, a huge number of combinations of route or link choices arise in some links. The problem is not only complex due to this fact, but also because each agent is trying to learn in this environment. This is a multiagent learning problem, for which we know there is no guarantee of convergence to the optimum choice of the users. This user equilibrium can be found only for very simple networks, and they consider aggregate flows. Section 2 discusses this and approximate methods. However, they are not efficient and cannot handle fully individual choices, thus they miss a significant portion of the space of combinations of route or link choices, eventually missing the optimal solution. Furthermore, classical methods do not handle en-route replanning, i.e., changes in the initially planned route during the actual trip.

In short, in this paper it is argued that a RL-based method at individual agent level, though not guaranteed finds the optimum solution for the trip assignment, has advantages over some classical methods. Perhaps the most important is that it allows a higher degree of heterogeneity in the choices of routes, without assuming a central authority that has global information, as it is the case with some classical trip assignment methods. The consequences of this is a better distribution of the road users in the road network. To illustrate this, the present paper uses a non-trivial scenario, in which some links are highly attractive to all agents, but produce severe congestion if all of them use those links in their trips.

This problem has not been adequately addressed in the literature. The traffic engineering literature mainly takes the aggregate perspective (see Section 2). When dealing with disaggregate modeling, it is not individual-based in the sense that each individual can make its own choice on link basis, as in the present paper. Rather, *portions* of the individuals make the same decision about which route to use. A coarse discretization has severe implications as discussed later. In the autonomous agents and multiagent systems literature, scenarios dealing with more than two or three routes, and those in which agents can change their routes on the fly are just beginning to be investigated. It is unclear what happens when drivers can adapt to traffic patterns in complex traffic networks. From the point of view of the whole system, the goal is to ensure reasonable travel times for all users, which can be conflicting with some individual utilities. Some of these works are discussed in Section 3.

Apart from background concepts, methods, and related work on trip assignment, which are discussed in the next two sections, Section 4 describes the proposed approach and the scenario used to illustrate it. Results are shown and analyzed in Section 5, while Section 6 presents the concluding remarks and points to future research.

## 2. TRAFFIC ASSIGNMENT METHODS

In this section, basic concepts about traffic (or trip) assignment methods are given. For an extensive explanation, please refer to Chapter 10 in [12] or to Chapter 4 in [3].

A road network can be represented as a graph $G = (V, E)$, where $V$ is the set of vertices that represent the intersections of the network, and $E$ is a set of directed arcs, describing the existing road segments as directed connections between pairs of vertices. Each link $l_k \in L$ has a cost $c_k$, which is given by a function that takes as input attributes such as length, toll, free-flow speed (and hence, free-flow travel time), capacity, current volume, etc. A route $r_p$ is defined by a set of connected nodes $(n_0, n_1, n_2, ...)$. The length of each $r_p$ is the sum of the lengths of all links $l_k$ that connect these nodes.

Another relevant concept that needs to be introduced here is the one of volume-delay functions (VDFs) or cost-flow relationship. These are used in macroscopic modeling and aim at accounting for congestion effects, i.e., how over-capacity of a given volume or flow in a link affects the speed and travel times (costs of delays). These functions account for the flows in the whole network, i.e., they consider the interactions between flows that use the network at the same time, and the corresponding delays that may occur. As a simple example of a VDF, one can consider the following: $t_k = t_{k_0} + 0.02 \times q_k$. Here, $t_k$ is the travel time on link $k$, $t_{k_0}$ is the travel time per unit of time under free flow conditions, and $q_k$ is the flow using link $k$. This means that the travel time in each link increases by 0.02 of a minute for each vehicle/hour of flow.

Given a demand $T_{ij}$ for trips between origin $i$ and destination $j$, there are several schemes to assign these trips to the links of a road network. Such schemes can be classified over two main dimensions: (1) are capacity constraints included?, and (2) are stochastic effects included? The classical scheme for situations in which there are no congestion effects and no stochasticity in route choices is the all-or-nothing scheme (discussed later). Stochasticity is handled by simulation-based methods. Assignment under congestion is of course a hot research topic and many approaches exist in this category. If one ignores the stochastic effects and focus on capacity constraints, the aforementioned concept of VDFs play a major role. For example, given VDFs for each link in the network, a goal of these approaches is to approximate the equilibrium conditions as stated by Wardrop [19]: "under equilibrium conditions traffic arranges itself in congested networks such that all used routes between an OD pair have equal and minimum costs while all those routes that were not used have greater or equal costs". This is Wardrop's first principle, also known as Wardrop's equilibrium or user equilibrium.

Thus, given a traffic network, the assignment from the point of view of the user equilibrium can be analytically stated as an optimization problem: find all flows from each OD pair s.t. only paths with minimal costs have a nonzero flow assigned to them, which corresponds to Wardrop's first principle. For a mathematical formulation of this problem, the reader is referred to Chapter 2 in [7], as well as to [14].

One problem with this scheme is that it is not possible to solve the equilibrium flows algebraically, except for very simple cases (e.g., two or three links connecting a single OD pair). Thus, approximate solutions to the Wardrop's equilibrium were proposed. To evaluate their quality, relevant issues are solution stability and convergence, as well as computational requirements.

Such approximate solutions are discussed later in this section. Before, it is important to introduce a general procedure that underlies any of the assignment schemes. Indeed, each assignment scheme discussed before has several steps that must be treated in turn: (i) to identify a set of routes that might be considered attractive to drivers; (ii) to assign suitable proportions of the trip matrix to these routes; this results in flows on the links of the network; (iii) to search for convergence: many techniques follow an iterative pattern of successive approximations to an ideal solution (e.g., Wardrop's equilibrium).

The first step can be accomplished with any variant of the Dijkstra algorithm for shortest paths. This step is also known as tree-building step. However, normally these paths are generated based on a first-approximation or an estimated cost function (e.g., one that considers no congestion, i.e., only free-flow travel times are considered) because the real cost is not known, given that it depends on the route choices of all users. Therefore, in a non-free-flow regime (i.e., under congestion), the second aforementioned step must be performed iteratively, until some sort of convergence is reached.

Next, some classical trip assignment approaches are discussed. The typical approach to trip assignment under no congestion is to assign all trips to the route with minimum cost, on the basis that these are the routes travelers would rationally select. That is as in Eq. 1, where $T_{ij}$ is the given demand between origin $i$ and destination $j$. This procedure is referred as "all-or-nothing" assignment. It is possible to see that this scheme assigns all trips between nodes $i$ and $j$ to the same links (because, as mentioned, this scheme assumes no congestion).

$$\left.\begin{array}{ll} T_{ijr^\star} = T_{ij} & \text{for the minimum cost route } r^\star \\ T_{ijr} = 0 & \text{for all other routes} \end{array}\right\} \quad \forall_{i,j} \quad (1)$$

For route assignment under congestion (i.e., the capacity of a link $k$ can be surpassed and, as such, a VDF is necessary to account for the effects of the over-capacity), mainly two iterative methods can be used. The first is to load the network incrementally in $n$ stages, e.g., assigning a given fraction $p_n$ (e.g., 10%, 20%, etc.) of the total demand (for each OD pair) at each stage. Further fractions are then assigned based on the newly computed link costs. This procedure continues until 100% of the demand is assigned. Typical values for fractions $p_n$ are 0.4, 0.3, 0.2, and 0.1. An algorithm for this is the following (adapted from [12]):

1. select an initial set of current link costs (usually the free-flow travel times); initialize flows at all links $k$: $V_k = 0$; select a fraction $p_n$ of the trip matrix $T$ such that $\sum_n p_n = 1$; make $n = 0$.
2. build the set of minimum cost trees (one for each origin) using the current costs; $n \leftarrow n + 1$.
3. load $T_n = p_n T$ all-or-nothing trips to these trees, obtaining a set of auxiliary flows $F_k$; accumulate flows on each link: $V_k^n = V_k^{n-1} + F_k$.
4. calculate a new set of current link costs based on flows $V_k^n$; if not all fractions of $T$ have been assigned, proceed to step 2.

It must be remarked that there is no guarantee that this algorithm converges to the Wardrop's equilibrium, no matter how small each $p_n$ is. This procedure has the drawback that once a flow has been assigned to a link, due to the accumulated nature (see step 3), it is never removed. Thus, in case an arbitrarily low over-capacity is assigned to a link, then it prevents the convergence to the optimum solution. However, it is very easy to program.

The other approach is to start from some initial values for the link costs and find the minimum cost routes. Trips are then assigned to these routes. New costs are computed and this cycle is repeated until there is no significant change in link or route volumes. For instance, in the method of successive averages, the flow at the n-th iteration is calculated as a linear combination of the flow on the previous iteration and an auxiliary flow resulting from an all-or-nothing assignment in the n-th iteration. This can be formalized as the following procedure (again, adapted from [12]):

1. select an initial set of current link costs (usually the free-flow travel times); initialize flows at all links $k$: $V_k = 0$; make $n = 0$.
2. build the set of minimum cost trees (one for each origin) using the current costs; $n \leftarrow n + 1$.
3. load the whole of the matrix $T$ all-or-nothing to these trees obtaining a set of auxiliary flows $F_k$.
4. calculate the current flows as: $V_k^n \leftarrow (1 - \phi)V_k^{n-1} + \phi F_k$, with $0 \leq \phi \leq 1$.
5. calculate a new set of current link costs based on $V_k^n$; if no $V_k^n$ has changed significantly in two consecutive iterations, stop; otherwise proceed to step 2 (or, alternatively, use a maximum number of iteration).

The last step of the method admits several ways to fix the value of $\phi$. A useful one is to make $\phi = 1/n$. There is a proof that this produces solutions convergent to the Wardrop's equilibrium but this may be very inefficient.

Note that both iterative methods to approximate Wardrop's equilibrium are based on the all-or-nothing scheme (applied in each iteration). Thus, even for fine discretization levels, a number of trips is assigned to the same links.

# 3. RELATED WORK

A number of works from transportation planning and economics, as well as from mathematics and operations research, physics, and computer science deal with this problem. Computer science plays a role when it comes to solving large-scale road network problems. The most relevant and close to the approach proposed here are discussed next.

In [11], the author makes the point that trip assignment schemes that are based on steady-state flow conditions of the road network are adequate only for analysis of long-term strategic planning horizons, but not for tactical measures that are of interest in applications around intelligent transportation systems. However, the author also recognizes the challenges of finding an analytic representation that satisfies the laws of physics and traffic sciences, while also being mathematically tractable. Therefore they propose a simulation-based approach for the dynamical traffic assignment (DTA) problem. In DTA one goal is to describe how flows develop not only spatially but also temporally in the network. This means that DTA considers road users that depart from an origin to a destination at different times. Hence, they experience different travel times and, as such, the user equilibrium condition applies only to travelers who are assumed to depart at the same time between the same OD pair. In the present work, departure at different times is not considered, but the approach is not purely simulation-based given that the agents learn by interacting with the environment. DTA is also the focus of [16] in which the authors propose a predictive DTA model, also based on simulation and combined with the method of successive averages. Henn ([8]) proposes a fuzzy-based method to take the imprecisions and the uncertainties of the road users into account. These predict costs for each path based on a fuzzy subset that can represent imprecision on network knowledge, as well as uncertainty on traffic conditions.

As mentioned, a natural way to represent the problem of route choice (in opposition to trip assignment) is to model it using an agent-based modeling and simulation approach. Hence, there has been some works in this direction. An example is MATSim [1, 2], which deals with activity-based simulation of route choice.

Route choice under various levels of information is turning a hot research topic due to the increasing use of navigation devices. Agent-based route choice simulation has been applied to research concerning the effects of intelligent traveler information systems. Main questions here are what happens to the overall demand, if a certain share of drivers is informed and adapt. What kind of information is the best one to be given? Examples for such research line can be found in [9] for a two-route scenario, or in [6] where a neural net-based agent model for route choice is presented regarding a three route scenario. In [13], a simple network for fuzzy-rule based routing (including qualitative decisions) is used.

One problem with these approaches is that their application in networks with more than a couple of routes between a few locations is not trivial. The first problem is that a set of reasonable route alternatives has to be generated. A $n$-shortest path algorithm can be used but it may output routes that differ only marginally. Additionally, all approaches, including agent-based ones, consider one route as one complete option to choose. On-the-fly re-routing has hardly been a topic for research. Even more sophisticated agent architectures such as the one proposed by [15] do not include the possibility of re-routing during the trip.

To address this issue, re-routing in a scenario with multiple origins and destinations was studied in [4]. Besides route choice by the driver agents, the authors also consider traffic lights as adaptive agents in order to test whether such a form of co-adaptation may result in interferences or positive cumulative effects. This was one of the first works in the agent-based community that has dealt with agents computing new routes on the fly. This is important because en-route modifications cannot be ignored in a realistic simulation of decision making in traffic. An abstract route choice scenario was used, having some features of real world networks. However, in this work no comparison is made to methods that approximate the user equilibrium thus, it is not possible to fully assess the quality of those results.

Degradation in performance caused by the selfish behavior of individual road users remains an important research topic. [10] have proposed the so-called price of anarchy to measure this degradation. They show results for small networks such as the one used to illustrate the Braess paradox.

A learning-based approach was used by [17] where agents learn to select routes; thus there is no en-route changes in the routes. The size and topology of the network is not mentioned but it seems to be a single origin and destination.

# 4. APPROACH AND CASE STUDY

One of the problems with the methods discussed in Section 2 is that the set of routes that are considered in each step of the iterative process is reduced in order to gain in terms of computing time. However, this set can be far from the set that would be used by real world drivers, even if considering their informational constraints regarding the status of the traffic at the moment they make decisions. In other words, the granularity of the route selections is very coarse.

The approach proposed here can handle much finer granularities; actually, there is no limitation or restriction on the number and kinds of routes that users can select. This means that all possible routes can be combined (one for each agent), contrarily to schemes discussed in Section 2. This occurs because the granularity of those schemes is coarse per se. For example, the all-or-nothing approach is the extreme case where the whole volume for each OD pair is assigned to the same route. However, even less coarse methods as for instance the incremental method, still assigns the same route to a given fraction of road users. Of course these fractions can be small but the efficiency of this method decreases with the discretization (number of incremental steps). Similarly, in the successive averages method, the computational cost of the method depends on a good choice of the parameter $\phi$. If it is a function of the parameter $n$ (see Section 2), then the efficiency may be compromised.

In the iterative methods it is not possible to actually assign a different route to each road user at each iteration, as the method proposed here does. For this, this method pays a cost (more iterations are necessary as agents are learning while selecting routes) but it is still a tractable method given that each iteration typically takes just a few minutes. As it will be discussed further, the proposed method and the iterative methods present basically similar running times, but the former is heterogeneous in terms of combinations of individual route choices, thus exploring the possible search space in ways that are not possible with the iterative methods (without incurring in much higher running times).

The approach proposed here is based on RL. Agents learn the value of their actions by interacting with an environment that gives a feedback signal to each agent, based on which state the agent is in, and the action this agent decides to make while in that state. RL problems can be modeled as Markov decision processes (MDPs). An experience tuple $\langle s, a, s', r\rangle$ denotes the fact that the agent was in state $s$, performed action $a$ and ended up in $s'$ with reward $r$. Here, a popular model-free algorithm for RL is used, namely Q-learning. The update rule for each experience tuple $\langle s, a, s', r\rangle$ is given in Equation 2, where $\alpha$ is the learning rate and $\gamma$ is the discount for future rewards.

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \, max_{a'} \, Q(s',a') - Q(s,a)\right) \quad (2)$$

Considering a high number of agents in multi-agent RL turns the problem inherently more complex. This complexity has many causes and consequences, one being that mathematical convergence guarantees no longer hold.

The learning agents are the road users; the environment is a road network, where nodes form the set of states an agent may be, and the links departing from each node form the set of actions an agent may take. Each agent has an origin and a destination. Routes connecting these two are represented as a set of consecutive nodes. Of course there are many ways in which a destination node can be reached. Because links have a travel time that depends on the number of agents using them in their route choices, this problem is complex. Mostly, the desirable, shortest path under free-flow, may end up producing a high travel time if too many agents want to use it. Agents then need to learn how go from their origins to their destinations by finding routes that allows them to distribute themselves in such a way that the optimal number of agents use each link, minimizing their travel times.

In order to address a non trivial network, the one suggested in [12] (Exercise 10.1) is used, as depicted in Figure 1. All links are two-way. This network represents two residential areas (nodes A and B) and two major shopping areas (nodes L and M). The numbers in the links are their travel times under free flow (in both ways). These also appear in the second column of Table 3. For the shortest path algo-

**Table 1: Shortest Paths and Free-Flow Travel Times (FFTT) for the Four OD Pairs (original and modified networks).**

| OD | original | | modified | |
|---|---|---|---|---|
| pair | sh.st path | FFTT | sh.st path | FFTT |
| AL | ACGJIL | 28 | ACDGJIL | 23 |
| AM | ACDHKM | 26 | ACDGJKM | 23 |
| BL | BDGJIL | 32 | BDGJIL | 22 |
| BM | BEHKM | 23 | BDGJKM | 22 |

rithms, these can be seen as their costs so henceforth both terms are used indistinctly.

Further, in order to make the assignment more complex, two modifications in the fixed costs were made: to make an arterial more attractive to all road users (and hence the learning effort more difficult as there is more competition for cheap resources), the fixed costs of links DG and GJ (and their opposite directions as well) were reduced from, respectively, 7 and 3 to zero. These modifications are indicated in Table 3 by shadowed cells. This way the shortest paths, for each OD pair, and their travel times (under free-flow) are shown in Table 1, both for the original network and for the modified network. Note that the proposed approach (as well as the iterative methods) were run for both the original and for the modified versions but since the latter is more challenging, only results steaming from the experiments using the latter are mentioned. Notice, however, that the general conclusions are valid for both, i.e., the RL-based approach outperforms the other methods. Henceforth this network is referred as OW network.

In the experiments, 1700 driver agents were used as this is the proposed demand for the OW network during a Saturday morning peak (see exercise 10.1 in the book) and an estimated demand from A and B to L and M as depicted in Table 2. Further, the exercise proposes a VDF that relates cost $c(q_k)$ at link $k$ to its flow $q_k$. Specifically, it is proposed that the travel time in each link is increased by 0.02 for each vehicle/hour of flow ($t_k = t_{k_0} + 0.02 \times q_k$, as discussed in Section 2).

This simple scenario goes far beyond simple two-route (binary) choice scenario. It captures properties of real-world scenarios, like interdependence of routes with shared links and heterogeneous capacities and demand throughout the complete network. Moreover, the number of possible routes between two locations is high and/or it may involve loops as links are two-way, and it has more than a single OD pair. Hence, it is hardly possible to compute the Wardrop's equilibrium algebraically.

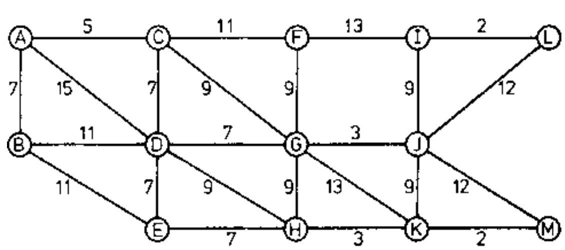



**Figure 1: Original Road Network (as proposed by Ortuzar and Willumsen)**

**Table 2: Average Travel Time per OD Pair: iterative methods**

| OD Pair | Trips | Incremental | Succ. Avgs. |
|---|---|---|---|
| AL | 600 | 69.00 | 68.04 |
| AM | 400 | 63.00 | 62.58 |
| BL | 300 | 69.60 | 64.50 |
| BM | 400 | 63.00 | 58.42 |
| | 1700 | 66.28 | 63.87 |

# 5. EXPERIMENTS AND RESULTS

Experiments were conducted using both the iterative methods discussed in Section 2, and the RL-based approach proposed here. As mentioned, the main aim is to show that a disaggregate, decentralized, agent-based approach in which agents learn by interacting with the environment, is able to find solutions that have at least the same travel times as the iterative methods, with little computational effort. Moreover, it is possible to show that the routes selected using the proposed approach are sometimes different from those found by the centralized, iterative approaches used as comparison. The fact that the RL-based approach was able to find lower travel times shows that the routes found by the iterative approaches could still be improved if more iterations were used, but this is unlikely to happen due to the coarse nature of discretization that underlies these methods.

Results for incremental and successive averages methods steam from the implementation of these algorithms provided by the publisher of Ortuzar and Willumsen's book. The shortest paths under free-flow (Table 1) were found algebraically. For the RL-based approach, simulations results reported here are averaged over 10 repetitions (for each condition). To render some tables cleaner, standard deviations are not always shown but they are of the order of 5% at most. Running times are in the order of few seconds for the iterative methods. For the RL-based approach, running times greatly depend on the number of episodes and on the value of the discount rate $\gamma$. For the cases shown next, simulations take at most a few minutes. Note however that the number of episodes can be greatly reduced, as indicated in the plots. Experiments were run in a standard PC (8 GB RAM), running Linux (for the RL-based approach) and Windows XP (for the algorithms provided by Ortuzar and Willumsen).

For evaluation, the performance measure is the same used in [12]: travel times averaged over all agents and also for each OD pair. Also, the number of trips using each link is shown, highlighting some differences found among the methods.

## 5.1 Results from Iterative Methods

Results for the iterative methods discussed before are shown in Tables 2 and 3; these were obtained using $p_n = 0.4, 0.3, 0.2, 0.1$ for the incremental method, and $\phi = 1/n$ and 100 iterations as stop criterion.

In Table 3, it is possible to see that both methods yield very different values for some links. Take links CD, JI, JK, JM, GK, KJ, DG, AD and BA as examples. Later, these numbers can be compared to the RL-based approach.

Table 2 summarizes the average travel times per OD pair for both methods, as well as the average travel times over all trips.

**Table 3: Travel Time Each Link: incremental and successive average methods.**

| | Fixed | Incremental | | Succ. Avg.s | |
|---|---|---|---|---|---|
| Link | Cost | Flow | Cost | Flow | Cost |
| AB | 7 | 0 | 7 | 4 | 7.08 |
| AC | 5 | 800 | 21 | 655 | 18.10 |
| AD | 15 | 200 | 19 | 348 | 21.96 |
| BA | 7 | 0 | 7 | 7 | 7.14 |
| BD | 11 | 370 | 18.40 | 374 | 18.48 |
| BE | 11 | 330 | 17.60 | 323 | 17.46 |
| CA | 5 | 0 | 5 | 0 | 5 |
| CD | 7 | 400 | 15 | 10 | 7.20 |
| CF | 11 | 240 | 15.80 | 372 | 18.44 |
| CG | 9 | 160 | 12.20 | 276 | 14.52 |
| DA | 15 | 0 | 15 | 0 | 15 |
| DB | 11 | 0 | 11 | 0 | 11 |
| DC | 7 | 0 | 7 | 3 | 7.06 |
| DE | 7 | 0 | 7 | 0 | 7 |
| DG | 0 | 770 | 15.40 | 551 | 11.02 |
| DH | 9 | 200 | 13 | 178 | 12.56 |
| EB | 11 | 0 | 11 | 0 | 11 |
| ED | 7 | 0 | 7 | 0 | 7 |
| EH | 7 | 330 | 13.60 | 323 | 13.46 |
| FC | 11 | 0 | 11 | 0 | 11 |
| FG | 9 | 0 | 9 | 0 | 9 |
| FI | 13 | 240 | 17.80 | 375 | 20.50 |
| GC | 9 | 0 | 9 | 0 | 9 |
| GD | 0 | 0 | 0 | 0 | 0 |
| GF | 9 | 0 | 9 | 3 | 9.06 |
| GH | 9 | 0 | 9 | 0 | 9 |
| GJ | 0 | 890 | 17.80 | 700 | 14 |
| GK | 13 | 40 | 13.80 | 124 | 15.48 |
| HD | 9 | 0 | 9 | 0 | 9 |
| HE | 7 | 0 | 7 | 0 | 7 |
| HG | 9 | 0 | 9 | 0 | 9 |
| HK | 3 | 530 | 13.60 | 501 | 13.02 |
| IF | 13 | 0 | 13 | 0 | 13 |
| IJ | 9 | 0 | 9 | 0 | 9 |
| IL | 2 | 600 | 14 | 450 | 11 |
| JG | 0 | 0 | 0 | 0 | 0 |
| JI | 9 | 360 | 16.20 | 75 | 10.50 |
| JL | 12 | 300 | 18 | 450 | 21 |
| JK | 9 | 320 | 15.40 | 8 | 9.16 |
| JM | 12 | 0 | 12 | 176 | 15.52 |
| KG | 13 | 0 | 13 | 0 | 13 |
| KH | 3 | 0 | 3 | 0 | 3 |
| KJ | 9 | 90 | 10.80 | 9 | 9.18 |
| KM | 2 | 800 | 18 | 624 | 14.48 |
| LI | 2 | 0 | 2 | 0 | 2 |
| LJ | 12 | 0 | 12 | 0 | 12 |
| MJ | 12 | 0 | 12 | 0 | 12 |
| MK | 2 | 0 | 2 | 0 | 2 |
| sum | | | 543.4 | | 522.38 |

## 5.2 Results of the RL based Approach

The approach presented in Section 4 has some parameters that refer basically to the Q-learning. These are the learning rate $\alpha$, the discount rate $\gamma$, and the exploration rate $\epsilon$. In the present paper, $\epsilon$ starts at $\epsilon_0 = 1$ and is multiplied by a factor of 0.995 at each episode in order to allow agents to explore the environment for a certain time. The value of this multiplicative factor must be set to fit the simulation horizon. As a general rule, 1000 episodes were run, so that that after 1000 episodes $\epsilon \approx 10^{-3}$. Notice that not all combinations of values for $\alpha$ and $\gamma$ require 1000 episodes. In some cases convergence to a given route choice pattern is reached much earlier, but for uniformity, the same number of episodes (1000) was used in all cases.

Next the results obtained when this approach was employed in the OW network are presented. Tables 4 and 5 show different measures that are of interest. First, Table 4 shows the average travel time over all 1700 trips, at the last episode, for different combinations of values for $\alpha$ and $\gamma$. To render it more clear, standard deviations are omitted and the numbers were rounded to integers. It is clear that the discount factor $\gamma$ plays a major role in the learning, while the learning rate $\alpha$ is less selective. This can be explained by the fact that choices that can be made at states that can be reached from a given state, are very important in this problem since the agent is trying to make a series of decisions in order to minimize travel time at the whole route. Therefore the discount rate must be high. It needs to be remarked that, in some cases, the number of trips over these links are much higher than the number of users. This is due to the fact that loops are possible and some users perform these loops. This is mainly the case when the discount rate is low and agents do not consider the future.

If one takes travel times given in Table 4, for different values of $\alpha$ and for the highest value of $\gamma$, it is is possible to see that these values (between 51 and 52) are lower than those shown in the last line of Table 2. Thus the RL-based approach yields travel times that are lower than the iterative methods, with roughly the same order of running times. Moreover, and perhaps more interesting, the choice made by the agents is based purely on local knowledge, whereas the iterative methods assume global knowledge of the links' costs.

Apart from values averaged over all links, it is interesting to check what happens in each link. As remarked before, the number of trips using some links differ much in the incremental and in the successive averages methods. Thus, a direct comparison with the RL-based approach is interesting. Table 5 shows the number of trips in selected links (to facilitate the comparison, numbers for the iterative methods were copied from Table 3), for $\alpha = 0.5$ and $\gamma = 0.99$. The criterion for inclusion in this table was that either the result achieved by the RL-based approach was different from both iterative methods, or it is close to one of these while these differ among them. For instance, for the link JM, the incremental and successive average methods assign zero and 176 trips respectively. The RL-based approach assigns 185 trips (with standard deviation – given the 10 runs – of about 8), thus being closer than the value obtained using the successive averages method. For cases in which the method proposed here differs from both, take for instance links AB and BA.

In this paper, an important point is that this difference, far from being bad, is what makes the RL-based approach more efficient. Links AB and BA were barely used in the trips assigned using the iterative methods. However, the learning agents found out that they can distribute themselves in ways that use the resources (links) in more efficient ways.

**Table 4: Average Travel Time (over all 1700 trips)**

| $\gamma$ | $\alpha$ | | | | |
|---|---|---|---|---|---|
| | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 |
| 0.99 | 52 | 52 | 51 | 51 | 51 |
| 0.8 | 50 | 50 | 50 | 50 | 50 |
| 0.6 | 58 | 56 | 55 | 55 | 57 |
| 0.5 | 84 | 80 | 81 | 76 | 80 |
| 0.4 | 114 | 111 | 100 | 102 | 107 |
| 0.2 | 329 | 225 | 183 | 152 | 181 |

**Table 5: Number of Trips Over Selected Links, for $\alpha = 0.5$ and $\gamma = 0.99$**

| Link | Incremental | Succ. Avg.s | RL-based: Avg. (Std. Dev.) |
|---|---|---|---|
| AB | 0 | 4 | 213 (8) |
| BA | 0 | 7 | 168 (4) |
| CD | 400 | 10 | 103 (5) |
| JM | 0 | 176 | 185 (8) |
| KJ | 90 | 9 | 5 (1) |

That travel times were efficient at user level was already discussed. A final comparison that can be made regards the sum of all costs, a measure of how efficient the method is at global level. The last line of Table 3 shows that the sum of costs over all links is over 500 for both iterative methods. When this sum is made considering the costs of links resulting from the RL-based approach, this value reaches only 462.94, with standard deviation of 0.22.

So far tables have shown the results of the assignment after 1000 learning episodes. The inset plot in Figure 2 depicts how the sum of links' costs change along time. The main plot shows how the number of trips changes with time, for three selected links: AB, BA and CD. These were selected because they show the greatest variation regarding the iterative methods, as shown in Table 5. Note that for $\alpha = 0.5$ and $\gamma = 0.99$, it would not be necessary to run 1000 episodes to reach convergence.

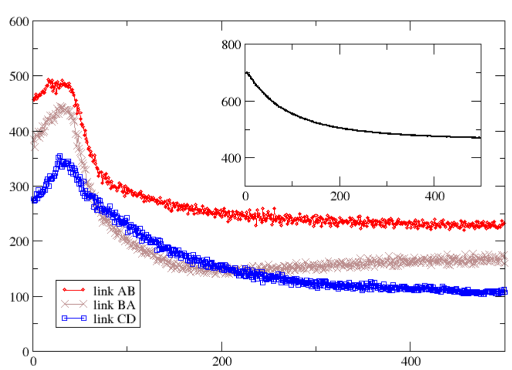



**Figure 2: Performance x time: sum of costs over all links (inset) and number of agents in selected links.**

## 6. CONCLUSIONS AND FUTURE WORK

Traffic assignment is an important step in modeling a transportation system. Classical approaches assume some degree of centralization, in which trips are assigned to links or routes. In this paper the perspective of the road user is taken: these users are modeled as agents that autonomously select their routes in an adaptive way. A similar perspective is taken in simulation-based works mentioned in Section 3, but there are two main differences to the present paper. First, here, agents do not anticipate traffic states (e.g., using fuzzy sets) but rather learn these states while interacting with the environment. This is a hard multiagent learning problem given the number of agents (here, thousands) trying to learn simultaneously in a competitive environment (links are shared by many agents). Second, agents form their routes while taking actions at nodes of the network, thus addressing the issue of en-route planning (as this task is known in traffic engineering, even if it is not a planning task from the AI point of view). In most previous simulation scenarios route adaptation was only allowed before and after the actual driving.

Results are twofold. First, the routes that are learned using the proposed approach are sometimes different from those found by the centralized, iterative approaches used as comparison. Second, the learning-based approach is more efficient than the iterative methods: exactly because the agents distribute themselves in different ways in the links of the road network (as compared to these approaches), the overall travel time is approximately 15% less than when iterative methods are used to assign trips to links. Also, the average travel time is lower for each of the OD pairs. This suggests that there is room for further improvements when the iterative methods are used. However, only few works reported in the literature show how far their results are from the optimum (only those dealing with simple networks).

A future direction of this work is to investigate the mathematical properties of the mathematical properties of the multiagent learning approach in order to provide insights about the bound to the optimum assignment. As this is a complex problem, one possibility is to use domain-dependent knowledge and/or properties of the domain. Also, a kind of reward shaping scheme as proposed in [18] can prove useful.

## Acknowledgments

Ana L. C. Bazzan is/was partially supported by CNPq and FAPERGS (grants 305062 and 10/0049-7 respectively). Gabriel Ramos has implemented the RL scheme for the original network design.

## 7. REFERENCES


[1] M. Balmer, N. Cetin, K. Nagel, and B. Raney. Towards truly agent-based traffic and mobility simulations. In N. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems, AAMAS*, volume 1, pages 60–67, New York, USA, July 2004. New York, IEEE Computer Society.

[2] M. Balmer, M. Rieser, K. Meister, D. Charypar, N. Lefebre, and K. Nagel. MATSim-T: Architecture and simulation times. In A. L. Bazzan and F. Klügl, editors, *Multi-Agent Systems for Traffic and*

*Transportation Engineering*, pages 57–78. IGI Global, Hershey, US, 2009.

[3] A. L. Bazzan and F. Klügl. Introduction to intelligent systems in traffic and transportation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 7(3):1–137, 2013.

[4] A. L. C. Bazzan and F. Klügl. Re-routing agents in an abstract traffic scenario. In G. Zaverucha and A. L. da Costa, editors, *Advances in artificial intelligence*, number 5249 in Lecture Notes in Artificial Intelligence, pages 63–72, Berlin, 2008. Springer-Verlag.

[5] M. Ben-Akiva and M. Bierlaire. Discrete choice methods and their applications to short term travel decisions. In *Handbook of transportation science*, pages 5–33. Springer US, 1999.

[6] H. Dia and S. Panwai. Modelling drivers' compliance and route choice behaviour in response to travel information. *Special issue on Modelling and Control of Intelligent Transportation Systems, Journal of Nonlinear Dynamics*, 49(4):493–509, 2007.

[7] C. Gawron. *Simulation-based traffic assignment*. PhD thesis, University of Cologne, Cologne, Germany, 1998.

[8] V. Henn. Fuzzy route choice model for traffic assignment. *Fuzzy Sets and Systems*, 116(1):77–101, 2000.

[9] F. Klügl and A. L. C. Bazzan. Route decision behaviour in a commuting scenario. *Journal of Artificial Societies and Social Simulation*, 7(1), 2004.

[10] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th annual conference on Theoretical aspects of computer science (STACS)*, pages 404–413, Berlin, Heidelberg, 1999. Springer-Verlag.

[11] H. S. Mahmassani. Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and Spatial Economics*, 1(3-4):267–292, 2001.

[12] J. Ortúzar and L. G. Willumsen. *Modelling Transport*. John Wiley & Sons, 3rd edition, 2001.

[13] S. Peeta and J. W. Yu. A hybrid model for driver route choice incorporating en-route attributes and real-time information effects. *Networks and Spatial Economics*, 5:21–40, 2005.

[14] B. Ran and D. E. Boyce. *Modeling dynamic transportation networks: an intelligent transportation system oriented approach*. Springer, 1996.

[15] R. Rossetti and R. Liu. A dynamic network simulation model based on multi-agent systems. In F. Klügl, A. L. C. Bazzan, and S. Ossowski, editors, *Applications of Agent Technology in Traffic and Transportation*, pages 88–93. Birkhäser, 2005.

[16] C. Tong and S. Wong. A predictive dynamic traffic assignment model in congested capacity-constrained road networks. *Transportation Research Part B: Methodological*, 34(8):625 – 644, 2000.

[17] K. Tumer and A. Agogino. Agent reward shaping for alleviating traffic congestion. In *Workshop on Agents in Traffic and Transportation*, Hakodate, Japan, 2006.

[18] K. Tumer, Z. T. Welch, and A. Agogino. Aligning social welfare and agent preferences to alleviate traffic congestion. In L. Padgham, D. Parkes, J. Müller, and S. Parsons, editors, *Proceedings of the 7th Int. Conference on Autonomous Agents and Multiagent Systems*, pages 655–662, Estoril, May 2008. IFAAMAS.

[19] J. G. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institute of Civil Engineers*, 1(3):325–362, 1952.

# Pedestrian Dynamics in Presence of Groups: an Agent-Based Model Applied to a Real World Case Study

Luca Crociani, Andrea Gorrini, Giuseppe Vizzari
Complex Systems and Artificial Intelligence research center,
Università degli Studi di Milano–Bicocca,
Viale Sarca 336/14, 20126 Milano, Italy
{name.surname}@disco.unimib.it

## ABSTRACT

The paper introduces an agent-based model for the simulation of crowds of pedestrians whose main innovative element is the representation and management of an important type of social interaction among the pedestrians: members of groups, in fact, carry out of a form of interaction (by means of verbal or non-verbal communication) that allows them to preserve the cohesion of the group even in particular conditions, such as counter flows, presence of obstacles or narrow passages. The paper formally describes the model and presents its application to a real world scenario in which an analysis of the impact of groups on the overall observed system dynamics was performed. The simulation results are compared to empirical data and they show that the introduced model is able to produce quantitatively plausible results in situations characterised by the presence of groups of pedestrians.

## Categories and Subject Descriptors

I.6 [**Simulation and Modeling**]: Applications

## General Terms

Experimentation

## Keywords

pedestrian and crowd modeling, interdisciplinary approaches

## 1. INTRODUCTION

The simulation of pedestrians and crowds is a consolidated and successful application of research results in the more general area of computer simulation of complex systems. Relevant contributions to this area come from disciplines ranging from physics and applied mathematics to computer science, often influenced by anthropological, psychological, sociological studies. The quality of the results provided by simulation models was sufficient to lead to the design and development of commercial software packages, offering useful functionalities to the end user (e.g. CAD integration, CAD-like functionalities, advanced visualisation and analysis tools) in addition to a simulation engine[1]. Pedestrian models can be roughly classified into three main categories that respectively consider pedestrians as *particles subject to forces*, particular *states of cells* in which the environment is subdivided in Cellular Automata (CA) approaches, or *autonomous agents* acting and interacting in an environment. The most widely adopted particle based approach is represented by the *social force model* [9], which implicitly employs fundamental proxemic concepts like the tendency of a pedestrian to stay away from other ones while moving towards his/her goal. *Cellular Automata* based approaches have also been successfully applied in this context: in particular, the floor-field model [5], in which the cells are endowed with a discretised gradient guiding pedestrians towards potential destinations. Finally, works like [10] essentially extend CA approaches, separating the pedestrians from the environment and granting them a behavioural specification that is generally more complex than what is generally represented in terms of a simple CA transition rule, but they essentially adopt similar methodologies. The resulting models are *agent–based*, since pedestrians are not merely states of cell. Relevant recent innovative studies employing agent-based approaches regard higher level aspects of pedestrian behaviour, like social aspects and the transfer of emotions in crowds (see, e.g., [4]) and they are not necessarily related to a discrete spatial representation of the simulated environment.

A recent survey of the field by [18] and by a report commissioned by the Cabinet Office by [6] made clear that, even after the substantial research that has been carried out in this area, there is still much room for innovations in models improving their performances both in terms of *effectiveness* in modelling pedestrians and crowd phenomena, in terms of *expressiveness* of the models (i.e. simplifying the modelling activity or introducing the possibility of representing phenomena that were still not considered by existing approaches), and in terms of *efficiency* of the simulation tools. Research on models able to represent and manage phenomena still not considered or properly managed is thus still lively and important. One of the aspects of crowds of pedestrians that has only been recently considered is represented by the implications of the presence of groups. A small number of recent works represent a relevant effort towards the modeling of groups, respectively in particle-based [15] (extending the social force model), in CA-based [17] (with adhoc approaches) and in agent-based approaches [16] (intro-

[1]See `http://www.evacmod.net/?q=node/5` for a large list of pedestrian simulation models and tools.

ducing specific behavioral rules for managing group oriented behaviors): in all these approaches, groups are modeled by means of additional contributions to the overall pedestrian behaviour representing the tendency to stay close to other group members. However, the above approaches only mostly deal with small groups in relatively low density conditions; those dealing with relatively large groups (tens of pedestrians) were not validated against real data. The last point is a crucial and critical element of this kind of research effort: computational models represent a way to formally and precisely define a computable form of theory of pedestrian and crowd dynamics. However, these theories must be validated employing field data, acquired by means of experiments and observations of the modeled phenomena, before the models can actually be used for sake of prediction. This paper represents a step in this direction, since it reports the results of a field observation and analysis of pedestrian and group behaviour (in the following section) then it introduces a model for pedestrian simulation encompassing an adaptive model for the preservation of group cohesion (Sect. 3) that is finally applied in a virtual counterpart of the observed scenario. Results of this simulation campaign are discussed in Sect. 4. Conclusions and future developments end the paper.

## 2. FIELD DATA ABOUT GROUPS

This Section comprises several empirical studies aimed at investigating pedestrian crowd dynamics in the natural context by using on-field observation. In particular the survey was aimed at studying the impact of grouping and proxemics behaviour on the whole crowd pedestrian dynamics. Data analyses were focused on: (i) *level of density and service*, (ii) *presence of groups* within the pedestrian flows, (iii) *trajectories and walking speed* of both singles and group members. Furthermore the *spatial dispersion* of group members while walking was measured in order to propose an innovative empirical contribution for a detailed description of group proxemics dynamics while walking.

The survey was performed the last 24$^{th}$ of November 2012 from about 2:50 pm to 4:10 pm. It consisted in the observation of the bidirectional pedestrian flows within the Vittorio Emanuele II gallery, a popular commercial-touristic walkway situated in the Milan city centre (Italy). The gallery was chosen as a crowded urban scenario, given the large amount of people that pass through it during the weekend for shopping, entertainment and visiting touristic-historical attractions in the centre of Milan.

The team performing the observation was composed of four people. Several preliminary inspections were performed to check the topographical features of the walkway. The balcony of the gallery, that surrounds the inside volume of the architecture from about ten meters in height, was chosen as location thanks to possibility to (i) position the equipment for video footages from a quasi-zenithal point of view and (ii) to avoid as much as possible to influence the behaviour of observed subjects, thanks to a railing of the balcony partly hiding the observation equipment. The equipment consisted of two professional full HD video cameras with tripods. The existing legislation about privacy was consulted and complied in order to exceed ethical issues about the privacy of the people recorded within the pedestrian flows.

Two independent coders performed a manual data analyses, in order to reduce errors by crosschecking their results. A square portion of the walkway was considered for data analysis: 12.8 meters wide and 12.8 meters long (163.84 square meters). In order to perform data analyses, the inner space of the selected area was discretised in cells by superimposing a grid[2] on the video (see Fig. 1); the grid was composed of 1024 squares 0.4 meters wide and 0.4 meters long. The video and the annotation data will soon be made available only for research purposes through the web.

### 2.1 Level of Density and Service

The bidirectional pedestrian flows (from North to South and vice versa) were manually counted minute by minute: 7773 people passed through the selected portion of the Vittorio Emanuale II Gallery from 2:50 pm to 4:08 pm. The average level of density within the selected area (defined as the quantitative relationship between a physical area and the number of people who occupy it) was detected considering 78 snapshots of video footages, randomly selected with a time interval of one minute. The observed average level of density was low (0.22 people/squared meter). Despite it was not possible to analyse continuous situations of high density, several situation of irregular and local distribution of high density were detected within the observed scenario.

According to the Highway Capacity Manual by [14], the level of density in motion situation was more properly estimated taking into account the bidirectional walkway level of service criteria: counting the number of people walking through a certain unit of space (meter) in a certain unit of time (minute). The average level of flow rate within the observed walkway scenario belongs to a $B$ level (7.78 ped/min/m) that is associated with an irregular flow in low-medium density condition.

### 2.2 Flow Composition

The second stage of data analysis was focused on the detection of groups within the pedestrian flows, the number of group members and the group proxemics spatial arrangement while walking. The identification of groups in the streaming of passerby was assessed on the basis of verbal and nonverbal communication among members: visual contact, body orientation, gesticulation and spatial cohesion among members. To more thoroughly evaluate all these indicators the coder was actually encouraged to rewind the video and take the necessary time to tell situations of simple local (in time an space) similar movements, due to the contextual situation, by different pedestrians from actual group situations. The whole video was sampled considering one minute every five: a subset of 15 minutes was extracted and 1645 pedestrians were counted (21.16% of the total bidirectional flows). Concerning the flow composition, 15.81% of the pedestrians arrived alone, while the 84.19% arrived in groups: 43.65% of groups were couples, 17.14% triples and 23.40% larger groups (composed of four or five members). Large structured groups, such as touristic committees, that were present in the observed situation, were analysed considering sub-groups.

[2]The grid was designed using *Photoshop CS5* (according to the perspective of the video images). An alphanumeric code was added on the sides of the grid. Finally, the grid with a transparent background was superimposed to a black-white version of the video images by means of *iMovie*. To perform counting activities, the video was reproduced by using *VLC* player thanks to its possibility to playback the images in slow motion and/or frame by frame and to use an extension time format that included hundredths of a second.

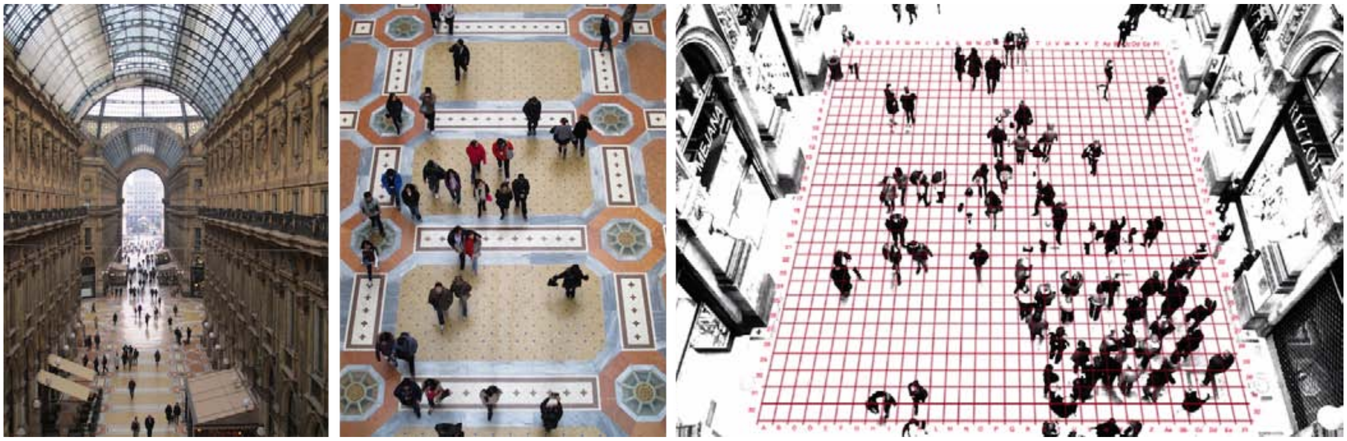


**Figure 1: From the left: an overview of the Vittorio Emanuele II gallery, the streaming of passerby within the walkway and a snapshot of the recorded video images with the superimposed grid for data analysis**

## 2.3 Trajectories and Walking Speed

The walking speed of both singles and group members was measured considering the path and the time to reach the ending point of their movement in the monitored area (corresponding to the centre of the cell of the last row of the grid) from the starting point (corresponding to the centre cell of the first row of the grid). Only the time distribution related to the *B* level of service was considered (as mentioned, the 59% of the whole video footages), in order to focus on pedestrian dynamics in situation of irregular flow. A sample of 122 people was randomly extracted: 30 singles, 15 couples, 10 triples and 8 groups of four members. The estimated age of pedestrians was approximately between 15 and 70; groups with accompanied children were not taken into account for data analyses. About gender, the sample was composed of 63 males (56% of the total) and 59 females (44% of the total). Differences in age and gender were not considered in this study. The selected pedestrians were chosen among those not stopping at shops' windows or entering shops, to actually focus on movement dynamics and not on the choice of activities (like in the vein of [8]).

The alphanumeric grid was used to track the trajectories of both single and group members within the walkway and to measure the length of their path[3] (considering the features of the cells: 0.4 m wide, 0.4 m long).

A first analysis was devoted to the identification of the length of the average walking path of singles (M=13.96 m, ±1.11), couples (M=13.39 m, ± 0.38), triples (M=13.34 m, ± 0.27) and groups of four members (M=13.16 m, ± 0.46). Then, the two tailed t-test analyses were used to identify differences in path among pedestrian. Results showed a significant difference in path length between: singles and couples (p value<0.05), singles and triples (p value<0.05), singles and groups of four members (p value<0.05). No significant differences were detected between path length of couples and triples (p value>0.05), triples and groups of four members (p value>0.05), couples and groups of four members (p value>0.05). The results showed that the path of singles is 4,48% longer than the average path of group members (including couples, triples and groups of four members).

The walking speed of both singles and group members was detected considering the path of each pedestrian within the flows and the time to reach the ending point from their starting point. A first analysis was devoted to the identification of the average walking speed of singles (M=1.22 m/s, ± 1.16), couples (M=0.92 m/s, ± 0.18), triples (M=0.73 m/s, ± 0.10) and groups of four members (M=0.65 m/s, ± 0.04). Then, the two tailed t-test analyses were used to identify differences in walking speed among pedestrian. Results showed a significant difference in walking speed between: singles and couples (p value<0.01), singles and triples (p value<0.01), singles and groups of four members (p value<0.01), couples and triples (p value<0.01), triples and groups of four members (p value< 0.05). In conclusion, the results showed that the average walking speed of group members (including couples, triples and groups of four members) is 37.21% lower than the walking speed of singles.

The correlated results about pedestrian path and speed showed that in situation of irregular flow singles tend to cross the space with more frequent changes of direction in order to maintain their velocity, avoiding perceived obstacles like slower pedestrians or groups. On the contrary, groups tend to have a more stable overall behaviour, adjusting their spatial arrangement and speed to face the contextual conditions of irregular flow: this is probably due to (i) the difficulty in coordinating an overall change of direction and (ii) the tendency to preserve the possibility of maintaining cohesion and communication among members.

## 2.4 Group Proxemics Dispersion

In order to improve the understanding of pedestrian proxemics behaviour the last part of the study is focused on the dynamic spatial dispersion of group members while walking. The dispersion among group members was measured as the summation of the distances between each pedestrian and the centroid (the geometrical centre of the group) all

[3]To measure the walking path and speed we considered each pedestrian as a point without mass in a two-dimensional plane. By using the alphanumeric grid, we considered the cell occupied by the feet of each pedestrian as its own actual position. The starting and final steps were measured from the half of the cell, consequently 0.2 m is the corresponding length of the each related path; any diagonal step cell by cell was measured as the diagonal between the two cells (0.56 m); any straight step was measured as the segment between the centre of two cells (0.4 m).

normalised by the cardinality of the group. The centroid was obtained as the arithmetic mean of all spatial positions of the group members, considering the alphanumeric grid. In order to find the spatial positions, the trajectories of the group members belonging to the previous described sample (15 couples, 10 triples and 8 groups of four members) were further analysed. In particular, the positions of the group members were detected analysing the recorded video images every 40 frames (the time interval between two frames corresponds to about 1.79 seconds, according to the quality and definition of the video images) starting from the co-presence of the all members on the alphanumeric grid. This kind of sampling permitted to consider 10 snapshots for each groups.

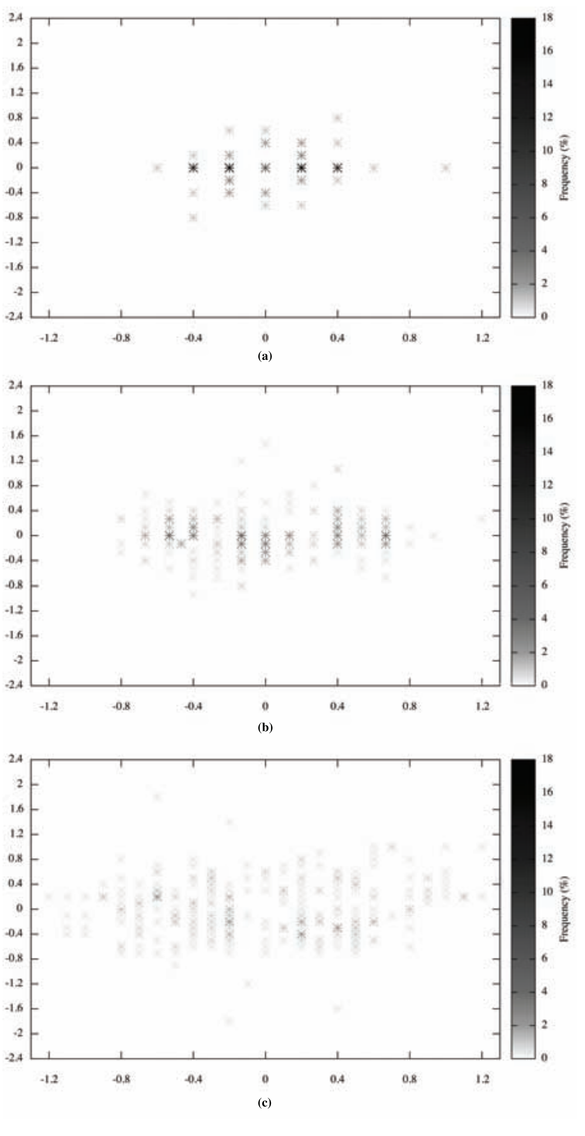



**Figure 2: A diagram showing most frequent positions, normalised with respect to the centroid and the movement direction, assumed by members of couples (a), triples (b) and groups of four members (c).**

A first analysis was devoted to the identification of the average proxemics dispersion of couples (M=0.35 m, ± 0.14), triples (M=0.53 m, ± 0.17) and groups of four members (M=0.67 m, ± 0.12). Then, the two tailed t-test analyses were used to identify differences in proxemics dispersion among couples, triples and groups of four members. Results showed a significant difference in spatial dispersion between: couples and triples (p value$<$0.05), couples and groups of four members (p value$<$0.01). No significant differences between triples and groups of four members (p value$>$ 0.05). In conclusion, the results showed that the average spatial dispersion of triples and groups of four members while walking is 40.97% higher than the dispersion of couples.

Starting from the achieved results about group proxemics dispersion, we finally focused on a quantitative and detailed description of group spatial layout while walking. The normalised positions of each pedestrian with respect to the centroid and the movement direction were detected by means of a sample of 10 snapshots for each groups (15 couples, 10 triple and 8 groups of four members) and then further analysed in order to identify the most frequent group proxemics spatial configurations, taking into account the degree of alignment of each pedestrian (see Figure 2). Result showed that couple members tend to walk side by side, aligned to the each other with a distance of 0.4 m (36% of the sample) or 0.8 m (24% of the sample), forming a line perpendicular to the walking direction (line abreast pattern); triples tend to walk with a line abreast layout (13% of the sample), with the members spaced of 0.60 m. Regarding groups of four members it was not possible to detect a typical spatial pattern: the reciprocal positions of group members appeared much more dispersed than in the case of smaller groups, probably to due the continuous arrangements in spatial positioning while walking.

## 3. PEDESTRIAN SIMULATION MODEL

In this section the formalisation of the agent-based computational model will be discussed, by focusing on the definition of its three main elements: *environment*, *update mechanism* and *pedestrian behaviour*.

### 3.1 Environment

The environment is modelled in a discrete way by representing it as a grid of 40 cm sided square cells size (according to the average area occupied by a pedestrian [20]). Cells have a state indicating the fact that they are vacant or occupied by obstacles or pedestrians: $State(c) : Cells \rightarrow \{Free,\ Obstacle,\ OnePed_i, TwoPeds_{ij}\}$.

The last two elements of the definition point out if the cell is occupied by one or two pedestrians respectively, with their own identifier: the second case is allowed only in a controlled way to simulate overcrowded situations, in which the density is higher than 6.25 pedestrians per square metre (i.e. the maximum density reachable by our discretisation).

The information related to the scenario[4] of the simulation are represented by means of *spatial markers*, special sets of cells that describe relevant elements in the environment. In particular, three kinds of spatial markers are defined: (i) *start* areas, that indicate the generation points of agents in the scenario. Agent generation can occur in *block*,

[4]It represents both the structure of the environment and all the information required for the realization of a specific simulation, such as crowd management demands (pedestrians generation profile, origin-destination matrices) and spatial constraints.

all at once, or according to a user defined *frequency*, along with information on type of agent to be generated and its destination and group membership; (ii) *destination* areas, which define the possible targets of the pedestrians in the environment; (iii) *obstacles*, that identify all the non-walkable areas as walls and zones where pedestrians can not enter.

Space annotation allows the definition of virtual grids of the environment, as containers of information for agents and their movement. In our model, we adopt the *floor field* approach [5], that is based on the generation of a set of superimposed grids (similar to the grid of the environment) starting from the information derived from spatial markers. Floor field values are spread on the grid as a gradient and they are used to support pedestrians in the navigation of the environment, representing their interactions with static object (i.e., destination areas and obstacles) or with other pedestrians. Moreover, floor fields can be *static* (created at the beginning and not changed during the simulation) or *dynamic* (updated during the simulation). Three kinds of floor fields are defined in our model: (i) *path field*, that indicates for every cell the distance from one destination area, acting as a potential field that drives pedestrians towards it (static). One path field for each destination point is generated in each scenario; (ii) *obstacles field*, that indicates for every cell the distance from neighbour obstacles or walls (static). Only one obstacles field is generated in each simulation scenario; (iii) *density field*, that indicates for each cell the pedestrian density in the surroundings at the current time-step (dynamic). Like the previous one, the density field is unique for each scenario.

Chessboard metric with $\sqrt{2}$ variation over corners [13] is used to produce the spreading of the information in the path and obstacle fields. Moreover, pedestrians cause a modification to the density field by adding a value $v = \frac{1}{d^2}$ to cells whose distance $d$ from their current position is below a given threshold. Agents are able to perceive floor fields values in their neighbourhood by means of a function $Val(f, c)$ ($f$ represents the field type and $c$ is the perceived cell). This approach to the definition of the objective part of the perception model moves the burden of its management from agents to the environment, which would need to monitor agents anyway in order to produce some of the simulation results.

## 3.2 Pedestrians and Movement

Formally, our agents are defined by the following triple: $Ped = \langle Id, Group, State\rangle$; where $State = \langle position, oldDir, Dest\rangle$, with their own numerical identifier, their group (if any) and their internal state, that defines the current position of the agent, the previous movement and the final destination, associated to the relative path field.

Before describing agent behavioural specification, it is necessary to introduce the formal representation of the nature and structure of the groups they can belong to, since this is an influential factor for movement decisions.

### 3.2.1 Social Interactions

To represent different types of relationships, two kinds of groups have been defined in the model: a *simple group* indicates a family or a restricted set of friends, or any other small assembly of persons in which there is a strong and simply recognisable cohesion; a *structured group* is generally a large one (e.g. team supporters or tourists in an organised tour), that shows a slight cohesion and a natural fragmentation into subgroups, sometimes simple.

Members of a simple group it is possible to identify an apparent tendency to stay close, in order to guarantee the possibility to perform interactions by means of verbal or non-verbal communication [7]. On the contrary, in large groups people are mostly linked by the sharing of a common goal, and the overall group tends to maintain only a weak compactness, with a following behaviour between members. In order to model these two typologies, the formal representation of a group is described by the following: $Group : \langle Id, [SubGroup_1, \ldots, SubGroup_m], [Ped_1, \cdots, Ped_n]\rangle$.

In particular, if the group is simple, it will have an empty set of subgroups, otherwise it will not contain any direct references to pedestrians inside it, which will be stored in the respective leafs of its three structure. Differences on the modelled behavioural mechanism in simple/structured groups will be analysed in the following section, with the description of the utility function.

### 3.2.2 Agent Behaviour

Agent behaviour in a single simulation turn is organised into four steps: *perception, utility calculation, action choice* and *movement*. The *perception* step provides to the agent all the information needed for choosing its destination cell. In particular, if an agent does not belong to a group (from here called *individual*), in this phase it will only extract values from the floor fields, while in the other case it will perceive also the positions of the other group members within a configurable distance, for the calculation of the *cohesion* parameter. The choice of each action is based on an utility value assigned to every possible movement according to the function $U(c) = \frac{\kappa_g G(c)+\kappa_{ob} Ob(c)+\kappa_s S(c)+\kappa_c C(c)+\kappa_i I(c)+\kappa_d D(c)+\kappa_{ov} Ov(c)}{d}$.

Function $U(c)$ takes into account the behavioural components considered relevant for pedestrian movement, each one is modelled by means of a function that returns values in range $[-1; +1]$, if it represents an *attractive* element (i.e. its goal), or in range $[-1; 0]$, if it represents a *repulsive* one for the agent. For each function a $\kappa$ coefficient has been introduced for its calibration: these coefficients, being also able to actually modulate tendencies based on objective information about agent's spatial context, complement the objective part of the perception model allowing agent heterogeneity. The purpose of the function denominator $d$ is to constrain the diagonal movements, in which the agents cover a greater distance ($0.4 * \sqrt{2}$ instead of 0.4) and assume higher speed respect with the non-diagonal ones.

The first three functions exploit information derived by local floor fields: $G(c)$ is associated to goal attraction whereas $Ob(c)$ and $S(c)$ respectively to geometric and social repulsion. Functions $C(c)$ and $I(c)$ are linear cobinations of the perceived positions of members of agent group (respectively simple and structured) in an extended neighbourhood; they compute the level of attractiveness of each neighbour cell, relating to group cohesion phenomenon. Finally, $D(c)$ adds a bonus to the utility of the cell next to the agent according to his/her previous direction (a sort of *inertia* factor), while $Ov(c)$ describes the *overlapping* mechanism, a method used to allow two pedestrians to temporarily occupy the same cell at the same step, to manage high-density situations. Overlapping plays an important role in preserving overall pedestrian flow in medium-high density situations (density higher than 2 pedestrians per square metre) [2].

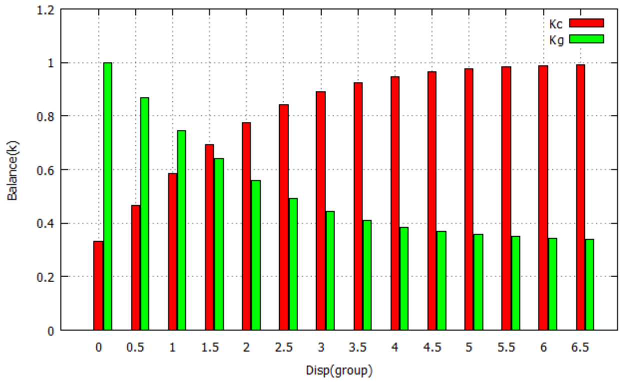



**Figure 3: Graphical representation of $Balance(k)$, for $k = 1$ and $\delta = 2.5$.**

As we previously said, the main difference between simple and structured groups resides in the cohesion intensity, which in the simple ones is significantly stronger. Functions $C(c)$ and $I(c)$ have been defined to correctly model this difference. Nonetheless, various preliminary tests on benchmark scenarios show us that, used singularly, function $C(c)$ is not able to reproduce realistic simulations. Human behaviour is, in fact, very complex and can react differently even in simple situation, for example by allowing temporary fragmentation of simple groups in front of several constraints (obstacles or opposite flows). Acting statically on the calibration weight, it is not possible to achieve this dynamic behaviour: with a small cohesion parameter several permanent fragmentations have been reproduced, while with an increase of it we obtained no group dispersions, but also an excessive and unrealistic compactness.

In order to face this issue, another function has been introduced in the model, to adaptively balance the calibration weight of the three attractive behavioural elements, depending on the fragmentation level of simple groups:

$$Balance(k) = \begin{cases} \frac{1}{3} \cdot k + (\frac{2}{3} \cdot k \cdot DispBalance) & \textit{if } k = k_c \\ \frac{1}{3} \cdot k + (\frac{2}{3} \cdot k \cdot (1 - DispBalance)) & \textit{if } k = k_g \vee k = k_i \\ k & \textit{otherwise} \end{cases}$$

where $DispBalance = tanh(\frac{Disp(Group)}{\delta})$, $Disp(Group) = \frac{Area(Group)}{|Group|}$, $k_i$, $k_g$ and $k_c$ are the weighted parameters of $U(c)$, $\delta$ is the calibration parameter of this mechanism and $Area(Group)$ calculates the area of the convex hull defined using positions of the group members. Fig. 3 exemplifies both the group dispersion computation and the effects of the $Balance$ function on parameters. The effective utility computation, therefore, employs calibration weights resulting from this computation, that allows achieving a dynamic and adaptive behaviour of groups: cohesion relaxes if members are sufficiently close to each other and it intensifies with the growth of dispersion.

After the utility evaluation for all the cells in the neighbourhood, the choice of action is stochastic, with the probability to move in each cell $c$ as ($N$ is the normalization factor): $P(c) = N \cdot e^{U(c)}$. On the basis of $P(c)$, agents move in the resulted cell according to their set of possible actions, defined as list of the eight possible movements in the Moore neighbourhood, plus the action to keep the position (indicated as $X$): $A = \{NW, N, NE, W, X, E, SW, S, SE\}$.

## 3.3 Time and Update Mechanism

In the basic model definition time is also discrete; in an initial definition of the duration of a time step was set to 0.31 $s$. This choice, considering the size of the cell (a square with 40 cm sides), generates a linear pedestrian speed of about 1.3 $m/s$, which is in line with the data from the literature representing observations of crowd in normal conditions [20], nonetheless we already implemented an extension of the model allowing the management of heterogeneous walking speeds [1].

Regarding the update mechanism, three different strategies are usually considered in this context [12]: *ordered sequential*, *shuffled sequential* and *parallel* update. The first two strategies are based on a sequential update of agents, respectively managed according to a *static* list of priorities that reflects their order of generation or a *dynamic* one, shuffled at each time step. The parallel update calculates instead the choice of movement of all the pedestrians at the same time, actuating choices and managing conflicts in a latter stage. The two sequential strategies imply a simpler operational management, due to an a-priori resolution of conflicts between pedestrians. For this work, we adopted the parallel update strategy, in accordance with the current literature, where it is considered much more realistic due to consideration of actual conflicts between pedestrians, arisen for the movement in a shared space [11].

With this update strategy, the agents life-cycle must consider that before carrying out the *movement* execution potential conflicts, essentially related to the simultaneous choice of two (or more) pedestrians to occupy the same cell, must be solved. The overall simulation step therefore follows a three step procedure: (i) *update of choices* and *conflicts detection* for each agent of the simulation; (ii) *conflicts resolution*, that is the resolution of the detected conflicts between agent intentions; (iii) *agents movement*, that is the update of agent positions exploiting the previous conflicts resolution, and *field update*, that is the computation of the new density field according to the updated positions of the agents.

The resolution of conflicts employs an approach essentially based on the one introduced in [11], based on the notion of friction. Let us first consider that conflicts can involve two of more pedestrians: in case more than two pedestrians involved in a conflict for the same cell, the first step of the management strategy is to block all but two of them, chosen randomly, reducing the problem to the case of a simple conflict among two pedestrians. To manage a simple conflict, another random number between 0 and 1 is generated and compared to two thresholds, $frict_l$ and $frict_h$, with $0 < frict_l < frict_h \leq 1$: the outcome can be that all agents are blocked when the extracted number is lower than $frict_l$, only one agent moves (chosen randomly) when the extracted number is between $frict_l$ and $frict_h$ included, or even two agents move when the number is higher than $frict_h$ (in this case pedestrian overlapping occurs). For our tests, the values of the thresholds make it quite relatively unlikely the resolution of a simple conflict with one agent moving and the other blocked, and much less likely their overlapping.

# 4. DISCUSSION OF TEST RESULTS

## 4.1 Configuration of the Simulation Scenario

| LoS | Size | Av. Dispersion | Observed |
|---|---|---|---|
| A | 2 | 0.336 (± 0.157) | |
| | 3 | 0.479 (± 0.153) | |
| | 4 | 0.575 (± 0.146) | |
| B | 2 | 0.351 (± 0.174) | 0.35 (± 0.14) |
| | 3 | 0.505 (± 0.194) | 0.53 (± 0.17) |
| | 4 | 0.609 (± 0.210) | 0.67 (± 0.12) |

**Table 1: Average groups dispersion achieved by the simulations (standard deviation inside breaks).**

| LoS | Size | Av. Speed | Observed |
|---|---|---|---|
| A | 1 | 1.19 | |
| | 2 | 1.115 | |
| | 3 | 1.119 | |
| | 4 | 1.11 | |
| B | 1 | 1.172 | 1.22 |
| | 2 | 1.107 | 0.92 |
| | 3 | 1.105 | 0.73 |
| | 4 | 1.099 | 0.65 |

**Table 2: Average speeds of groups.**

The environment is a discrete representation of the part of Vittorio Emanuele Gallery considered for the data extraction (see Sec. 2). It consists in a large corridor with size 12.8 m × 13.6 m. At each end, one *start area* is placed for the agents generation, respecting the frequency of arrival observed in the videos; corridor ends also comprise a destination area corresponding to the start area positioned on the other end. We decided not to consider the attractiveness of shops for two reasons: first of all, several shops in this section of the gallery are restaurants and they are not really much considered at the time of the observation, second the pedestrians selected for manual analysis did not stop at any shopping window. In order to reproduce the levels of service A and B we configured two different frequency profiles which lead to achieve, respectively, 30 and 50 pedestrians in the environment on average. The agent population comprises 15.8 % of individuals, while the remaining part is divided in groups of 2 (52%), 3 (20%) and 4 (28%) members, consistently with the observed composition of pedestrian population. In order to overcome biases caused by the simulation initialisation, for both density configurations, a set of 5 relatively short simulations (5 minutes of simulated time, for a total of 25 minutes of simulated time for both LOS conditions) has been run with different random seeds. Finally, simulations have been ran with the following configuration of the calibration weights: (i) utility function weights: $k_g = 8$, $k_{ob} = 2$, $k_s = 30$, $k_c = 6$, $k_I = 6$, $k_d = 2$ and $k_{ov} = 5$; (ii) cohesion mechanism: $\delta = 3.0$; (iii) friction weights: $frict_l = 0.8$, $frict_h = 0.96$.

## 4.2 Results

The data that can be gathered by means of a simulation covers a wide array of observable measurements. By means of this set of experiments, however, out main goal is the validation of reproduced behaviour of the agents inside groups, evaluating therefore the plausibility of the cohesion mechanism encompassed by the model.

The first measured data represents an indicator of the average dispersion of the different types of group during the simulation. Several methods have been proposed in the literature for describing the dispersion, since it is an intuitive concept that can however be formalised in different ways [3]. The results shown in Table 4.2 have been achieved by using the centroid method, describing dispersion as the average distance assumed by members of the group from its center of gravity, calculated as the average position of all the group members.

The results show that the cohesion mechanism is quite effective: the dispersion of groups in the two settings (LoS A and B) is similar and the increase of density have led to a very light increase of the average and standard deviation. The most important consideration, however, is the fact that these data are consistent with the empirically observed values (which refer to the B LOS conditions).

Table 4.2 shows instead the average speed characterising the movement of the different types of pedestrians (individuals or members of a certain type of group), calculated using the length of the actual trajectory and the time needed to move from the start area to any cell of the destination area of the corridor. In this case the model has only been able to reproduce results similar to the empirically observed data only for individuals and it only showed a slight decrease in the velocity of group members. On the other hand, it must be noted that all the agents have been configured with the same desired speed of 1.3 $m/s$, that is based on empirically observed velocity for pedestrians traveling for business purpose [19]. The same observation reports that pedestrians moving for leisure generally have a lower average walking speed. Therefore, our conjecture is that the much lower walking speed of groups might be due not only to the fact that members try to preserve the possibility to establish verbal and non-verbal communication, but also to a change in the reason and motivation for moving in the environment. Further analyses on this issue are object of future studies.

Finally, Table 4.2 analyses average travel distances covered by pedestrians in the simulations. This measure is obviously strictly related to the previous one, being actually used in the computation of the walking speed. As a consequence, even if simulated trajectories are very close to the measured ones also in this case the model was not able to differentiate paths covered by individuals and group ones (in some cases the traveled distance of individuals was actually lower, unlike in the observed data). In addition to the above considerations on motivations of the movement, that can also have an influence in the frequency of direction changes, we want to emphasise that a discrete model has intrinsic limits in the faithful reproduction of trajectories (that are inherently jagged and not as smooth as the real ones), so it could be difficult improving this kind of result adopting a discrete model.

# 5. CONCLUSIONS

This paper has introduced an empirical investigation of the influence of group presence in crowds of pedestrians by means of a field observation and a simulation campaign employing an agent-based model encompassing a specific adaptive mechanism for group behaviour in the same scenario. Empirical results achieved by the model are in tune with the

| LoS | Size | Av. Distance | Observed |
|---|---|---|---|
| A | 1 | 13.767 (± 0.57) | |
| | 2 | 13.922 (± 0.621) | |
| | 3 | 13.980 (± 0.624) | |
| | 4 | 14.052 (± 0.653) | |
| B | 1 | 13.857 (± 0.577) | 13.96 (± 1.11) |
| | 2 | 13.99 (± 0.628) | 13.39 (± 0.38) |
| | 3 | 14.049 (± 0.654) | 13.34 (± 0.27) |
| | 4 | 14.087 (± 0.653) | 13.16 (± 0.46) |

**Table 3: Average travelled distance of groups.**

actual observed data for the metrics related to group dispersion and walking trajectories. Nonetheless, additional work must be conducted to further understand if the lower speed of members of large groups is solely due to group influence or also to a change in the motivations of pedestrians, and therefore on their desired walking speed. Analogous considerations can be done for pedestrian trajectories although, in this case, the model is probably close to the intrinsic limits in the reproduction of smooth paths of any discrete approach.

# 6. REFERENCES

[1] S. Bandini, L. Crociani, and G. Vizzari. Heterogeneous speed profiles in discrete models for pedestrian simulation. In *Proceedings of the 93rd Transportation Research Board annual meeting*, 2014.

[2] S. Bandini, M. Mondini, and G. Vizzari. Modelling negative interactions among pedestrians in high density situations. *Transportation Research Part C: Emerging Technologies*, 40:251 – 270, 2014.

[3] S. Bandini, F. Rubagotti, G. Vizzari, and K. Shimura. An agent model of pedestrian and group dynamics: Experiments on group cohesion. In *AI*IA*, volume 6934 of *Lecture Notes in Computer Science*, pages 104–116. Springer, 2011.

[4] T. Bosse, M. Hoogendoorn, M. C. A. Klein, J. Treur, C. N. van der Wal, and A. van Wissen. Modelling collective decision making in groups and crowds: Integrating social contagion and interacting emotions, beliefs and intentions. *Autonomous Agents and Multi-Agent Systems*, 27(1):52–84, 2013.

[5] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3 - 4):507 – 525, 2001.

[6] R. Challenger, C. W. Clegg, and M. A. Robinson. Understanding crowd behaviours: Supporting evidence. Technical report, University of Leeds, 2009.

[7] M. Costa. Interpersonal distances in group walking. *Journal of Nonverbal Behavior*, 34:15–26, 2010.

[8] J. Dijkstra, J. Jessurun, H. J. P. Timmermans, and B. de Vries. A framework for processing agent-based pedestrian activity simulations in shopping environments. *Cybernetics and Systems*, 42(7):526–545, 2011.

[9] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51(5):4282–4286, May 1995.

[10] C. M. Henein and T. White. Agent-based modelling of forces in crowds. In *Multi-Agent and Multi-Agent-Based Simulation, MABS 2004*, volume 3415 of *Lecture Notes in Computer Science*, pages 173–184. Springer–Verlag, 2005.

[11] A. Kirchner, K. Nishinari, and A. Schadschneider. Friction effects and clogging in a cellular automaton model for pedestrian dynamics. *Phys. Rev. E*, 67:056122, May 2003.

[12] H. Klüpfel. *A Cellular Automaton Model for Crowd Movement and Egress Simulation*. PhD thesis, University Duisburg-Essen, 2003.

[13] T. Kretz, C. Bönisch, and P. Vortisch. Comparison of various methods for the calculation of the distance potential field. In *Pedestrian and Evacuation Dynamics 2008*, pages 335–346. Springer Berlin Heidelberg, 2010.

[14] J. S. Milazzo II, N. M. Rouphail, J. E. Hummer, and D. P. Allen. Quality of service for interrupted-flow pedestrian facilities in highway capacity manual 2000. *Transportation Research Record*, 1678(1):25–31, 1999.

[15] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS ONE*, 5(4):e10047, 04 2010.

[16] R. A. Rodrigues, A. de Lima Bicho, M. Paravisi, C. R. Jung, L. P. Magalhães, and S. R. Musse. An interactive model for steering behaviors of groups of characters. *Applied Artificial Intelligence*, 24(6):594–616, 2010.

[17] S. Sarmady, F. Haron, and A. Z. H. Talib. Modeling groups of pedestrians in least effort crowd movements using cellular automata. In *Asia International Conference on Modelling and Simulation*, pages 520–525. IEEE Computer Society, 2009.

[18] A. Schadschneider, W. Klingsch, H. Klüpfel, T. Kretz, C. Rogsch, and A. Seyfried. Evacuation dynamics: Empirical results, modeling and applications. In *Encyclopedia of Complexity and Systems Science*, pages 3142–3176. Springer, 2009.

[19] M. Schultz, C. Schulz, and H. Fricke. Passenger dynamics at airport terminal environment. In *Pedestrian and Evacuation Dynamics 2008*, pages 381–396. Springer Berlin Heidelberg, 2010.

[20] U. Weidmann. Transporttechnik der fussgänger - transporttechnische eigenschaftendes fussgängerverkehrs (literaturstudie). Literature Research 90, Institut füer Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau IVT an der ETH Zürich, 1993.

# Simulating Autonomous Pedestrians Navigation : A Generic Multi-Agent Model to Couple Individual and Collective Dynamics

Patrick Simo Kanmeugne
Université Pierre et Marie Curie
Laboratoire d'Informatique de Paris 6
patrick.simo-kanmeugne(at)lip6.fr

Aurélie Beynier
Université Pierre et Marie Curie
Laboratoire d'Informatique de Paris 6
aurelie.beynier(at)lip6.fr

## ABSTRACT

In this paper, we focus on planning credible walking paths in real-time for a potentially highly congested crowd of autonomous pedestrians. For this purpose, we exploit the principle of least effort, applied to human navigation, which postulates that credible behaviours emerge as a function of the organism's propensity to minimize metabolic energy expenditure with respect to task, environment dynamics, and organism's constraints to action [17]. We therefore propose a consistent problem formulation for the navigation task where both individual and collective dynamics are taken into account. Each pedestrian is represented as a situated agent who tries to reach its destination by following energy efficient paths. Agents are autonomous, and at the same time, subject to the environment dynamics. They interact with each other through the environment in order to estimate their energy expenditure relatively to their tasks. Our formulation results in a generic and scalable multi-agent model, capable of simulating individual and collective behaviours regardless of the number of agents.

## Keywords

pedestrian navigation, multi-agent simulation, interaction, coordination, traffic.

## 1. INTRODUCTION

Real-time pedestrian crowds simulation is a complex task for computer scientists. On the one hand, social studies on pedestrians' behaviours show that each pedestrian in a crowd behaves autonomously, conscientiously interacting with other pedestrians, while pursuing its own objectives [4]. On the other hand, empirical observations of pedestrians' flow in highly congested areas demonstrate some striking similarities between pedestrians' behaviours and particle flow dynamics [7].

Consequently to these apparently contradictory issues, designing philosophies diverge on whether to consider pedestrians' characteristics and local interactions, or to focus on pedestrians' flow regardless of individual characteristics, in order to formulate the underlying modelling principles. In the current literature, a naïve application of each of these philosophies is proved to lead to partially satisfying results. The first one could lead to intractable principles [9], resulting into models that struggle to reproduce collective behaviours like the edge effect [23] or the fingering effect [28]. The second philosophy could be inappropriate for low-density crowds, since it neglects pedestrian individualities, and might result into models that produce non-realistic individual behaviours [5].

In this paper, we explore the principle of least effort (PLE) [29] applied to human navigation, for a more generic approach. According to this principle, credible walking paths emerge as a function of the organism's propensity to minimize metabolic energy expenditure with respect to task, environment dynamics, and organism's constraints to action [17]. Several psychological studies on human movement showed that metabolic energy expenditure regulation is critical enough to explain both individual and collective behaviours among human beings [10, 29, 22, 17]. Following this idea, our contribution is:

1. a **consistent problem formulation** of the navigation task of autonomous pedestrians, where both individual and collective dynamics are taken into account. Pedestrians are situated agents who try to follow energy efficient paths towards their destinations. They use navigable resources, which recover the entire navigable space, to build their paths. Navigable resources mediates interaction between agents and provide dynamic measures that help the agents to estimate their energy expenditure relatively to their task.

2. a **generic multi-agent model**, in respect with our formulation, to perform real-time simulations of a potentially highly congested crowd. We will see that the *environment* concept from the multi-agent paradigm is particularly useful to tackle the complexity of the navigation task. The environment could be seen as an independent component that maintains dynamic measures used by agents to compute energy efficient paths. Moreover, since agents are situated in the environment and subjected to physical and dynamical constraints, an important part of the simulation dynamics could be delegated to the environment without compromising agents' autonomy.

Our work is close to IRM4S [15], continuum crowd [24] and PLEdestrian [5]. We use the same action theory as developed in the IRM4S model [15] and we adapt the agent model in order to fit the specificities of pedestrian navigation. Like Continuum crowd [24] and PLEdestrian [5], we use a least effort approach to model agents' behaviours and store dynamic information in the environment. However, the difference with our approach is that we specifically design the environment as a full component of the model, with a dedicated dynamics, different from that of agents.

We evaluate our work by submitting an online interview with videos of our model running on different low-density scenarios. We also run our model against some well known collective phenomenon – edge effect and fingering effect – with encouraging results in terms of credibility and scalability.

The rest of the document is organized as follows. In Section 2, we present related works on real-time pedestrians crowd simulation. In Section 3, we introduce our formulation of the navigation task for autonomous pedestrians. We also give an overview of the global architecture of our generic multi-agent model, and describe the role of each component. Sections 4 and 5 are respectively dedicated to the evaluation and the perspectives of our work.

## 2. RELATED WORK

Pedestrians crowd simulation is often tackled by using two types of approaches [21]: *microscopic* and *macroscopic* models.

Microscopic models are built upon pedestrians individual characteristics and local interactions, assuming that the combination of local interactions between agents – namely, collision avoidance mechanisms – and path following techniques, will result in the desired behaviour of the crowd. Helbing and Molnar [6] introduced the social force model (SFM) where each pedestrian is subjected to attractive or repulsive forces. For example, an attractive force could guide pedestrians toward their objectives, while a repulsive force keeps them away from obstacle or other pedestrians. The pedestrian dynamics is assumed to obey conservation laws, which leads to interesting collective behaviours. Reynolds [19] developed the concept of steering forces which are guiding forces that correspond to a pedestrian's preferences. For instance, a steering force could model the need to reach a predefined destination, to stay away from a given agent, or to stay close to a leading agent. Here, the agents dynamics do not obey any conservation laws, but the application of steering forces is ruled by a decisional architecture which is specific to each agent. The steering force paradigm is flexible and provides believable real-time animation [18]. Fiorini and Shiller [2] introduced the velocity obstacle paradigm which reduces the navigation problem of a mobile entity to the computation of an avoidance manoeuvre that ensures a collision-free navigation in a dynamic environment. Van den Berg et al. [26, 25] applied this paradigm and provided the RVO – Reciprocal Velocity Obstacle – model which is a robust adaptation for pedestrians real-time navigation.

One of the main challenge for microscopic model is the management of congestion [9]. Congestion management is more complex than simple collision avoidance since it involves both time and space considerations. It is also very critical because it influences the emergence of collective behaviours. To handle navigation in a congested area, microscopic models are often combined with global path planning techniques or mobile perception fields that helps the agent to perceive the dynamic features of the environment. Karamouzas et al. [13] used a dynamic uniform grid and couple a collision avoidance model with $A^*$ path-planning techniques. The dynamic grid provides density occupation insights to the agents who can, therefore, plan to avoid occupied areas. Saboia et al. [20] modified the SFM model to introduce a mobile grid attached to each agent. The mobile grid allows the agent to change its desired velocity at reasonable time and to navigate through congested areas. Similar techniques could be found in [11].

Undoubtedly, using dynamic information on the environment density and fast global path-planning technique *speeds-up* the simulation – when an agent avoids occupied areas, this automatically reduces the calls to a collision avoidance algorithm, which is the most expensive operation in such simulations. Nevertheless, most microscopic models separate local collision avoidance from global path planning, and conflicts inevitably arise between these two competing goals. Those conflicts tend to be exacerbated in highly congested areas or highly dynamic environments [24].

Macroscopic models offer a more objective modelling framework, concerning these last issues, by representing the crowd as a whole. Hughes [9] investigated the analytic properties of human flow and propose the following hypothesis to define a continuous human flow model:

1. The walking speed of pedestrians is determined by the density of surrounding pedestrians, the behavioural characteristics of the pedestrians, and the ground on which they walk.
2. Pedestrians have a common sense (potential) of the task they face to reach their common destination, such that any two individuals at different locations having the same potential would see no advantage to exchange their locations.
3. Pedestrians seek to minimize their estimated travel time but temper this behaviour to avoid extreme densities.

Treuille et al. [24] managed the resulting equations for real time simulation and define a *dynamic potential function* to formalize the navigation as an optimization problem. The resulting potential function is exploited to generate a dynamic vector field that governs the overall crowd behaviour.

Obviously, the underlying principles of a macroscopic models leave little room for agents' autonomy. Myopic collision avoidance behaviours and difficulty to handle several agents with different destinations, are among the most relevant drawbacks of such approaches. Nonetheless, it is also obvious that those models produce much more believable collective behaviours for highly congested crowd. We argue that those good performances are due to a more coherent optimization framework. We believe that it is possible to reproduce a similar framework while preserving pedestrians autonomy. Thus, we propose a new framework that uses the multi-agent paradigm, and develop a consistent formulation of the navigation task for autonomous pedestrians.

## 3. COUPLING INDIVIDUAL AND COLLECTIVE DYNAMICS

In this section, we present the formulation of the navigation task for autonomous pedestrians and the full specification of our multi-agent model.

## 3.1 Formulation

Inspired by the work of Whittle [27], Guy et al. [5] explicitly formulated the metabolic energy spent by pedestrians when they walk:

$$E = mass \cdot \int (e_s + e_w \cdot |v|^2) \cdot dt \qquad (1)$$

Where:

- $v$ is the pedestrian's instantaneous velocity
- $e_s$ and $e_w$ are individual attributes, respectively equal to $2.23 \frac{J}{kg \cdot s}$ and $1.26 \frac{J \cdot s}{kg \cdot m^2}$ for an average human[1]
- *mass* is the pedestrian's mass.

Kapadia et al. [12] extended this formula to include a specific *collision effort* which is the amount of energy that is expended through collisions:

$$E = mass \cdot \int (e_s + e_w \cdot |v|^2 + e_c \cdot c_p(t)) \cdot dt \qquad (2)$$

Where:

- $c_p(t)$ estimates the penetration depth of the collision if the agent is colliding with another agent at that point of time.
- $e_c = 10 \frac{J}{kg \cdot m \cdot s}$ is a penalty constant for collisions.

We formulate the navigation task of our agents in regards to this last equation: each agent will try to minimize it, individually and subjectively, by speculating on its surrounding's dynamics and adapting its walking behaviour accordingly.

### 3.1.1 Resources and Task

To support the navigation task, we assume a 2D continuous space which is discretized into contiguous triangular meshes of homogeneous size. Each triangular mesh is a *navigable resource* that will be used by agents to build their path.

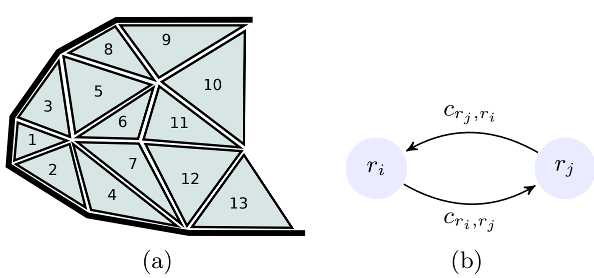



**Figure 1: Structure of the continuous space. Each resource $r_i$ is materialized by a triangular mesh. Each pair of resources $(r_i, r_j)$ represents a discretized movement from $r_i$ to $r_j$. The real-time cost of a discretized movement $(r_i, r_j)$ is noted $c_{r_i,r_j}$.**

Figure 1(a) gives an overview of the topological structure extracted from the continuous space. We choose triangular meshes because they allow us to recover the entire space with no discontinuities.

[1] J: Joules; kg: kilograms; $m^2$ : square meters

Our topological structure induces an oriented graph where each resource $r_i$ represents a node, and each pair of contiguous resources $(r_i, r_j)$ represents a *discretized movement*. We associate a real time cost $c_{r_i,r_j}$ to each discretized movement $(r_i, r_j)$, to be valued relatively to an agent: $c_{r_i,r_j}$, at a given time $t$, represents the average metabolic energy that the agent expects to spend if it travels from $r_i$ to $r_j$ at $t$ (Figure 1(b)).

Consequently, a first formulation of the navigation task could be stated as: following the energy most efficient path available from a given position $A$ towards a destination $B$, where the path is represented as a suite of contiguous resources $(r_i)_{1 \leq i \leq k}$, and the total energy of a path is estimated as the overall cost of the discretized movements that constitute it. This corresponds to the following decision problem:

$$\begin{cases} \text{find } (r_i)_{1 \leq i \leq k} \text{ such as} \\ A \in r_1 \\ B \in r_k \\ r_i \text{ and } r_{i-1} \text{ are contiguous } \forall i > 1 \\ \min \sum_{i=2}^{i=k} c_{r_{i-1},r_i} \end{cases} \qquad (3)$$

With,

$$\begin{aligned} c_{r_{i-1},r_i} = \; & e_s \cdot D_{r_{i-1},r_i} + \\ & e_w \cdot S_{r_{i-1},r_i} \cdot |v_{r_i,r_{i-1}}|^2 + \\ & e_c \cdot D_{r_{i-1},r_i} \cdot (q_{r_i} + q_{r_j}) \end{aligned} \qquad (4)$$

Where,

- $D_{r_i,r_j}$ is a real time estimation of the mean total travel time from $r_i$ to $r_j$ including the potential delays due to congestion.
- $S_{r_i,r_j}$ is a real time estimation of the mean travel time from $r_i$ to $r_j$ excluding the delays due to congestion.
- $v_{r_i,r_j}$ is a real time estimation of the mean travel speed from $r_i$ to $r_j$.
- $q_{r_i}$ is a real time estimation of the mean number of agents in $r_i$.

Equation (4) corresponds to our estimation of the metabolic energy expenditure for a discretized movement, drawn from equation (2). As stated above, it represents the real time estimation of the amount of energy that the agent expects to spend if it travels from $r_i$ to $r_j$. Here, we suggest that the number of expected collisions is proportional to the mean number of agents in both resources $r_j$ and $r_j$. For simplicity, we have neglected the contribution of the mass and considered a constant penetration depth for collisions.

$D_{r_i,r_j}$, $S_{r_i,r_j}$, $q_{r_i}$, $q_{r_j}$ and $v_{r_i,r_j}$ are stochastic measures that are estimated relatively to an agent. Since they are closely related to the traffic, we also call them *dynamic information variables*. In the next section, we propose an explicit formulation of those variables. For that purpose, we introduce an independent traffic module which is in charge of converting the collective dynamics into individual utilities.

### 3.1.2 Converting Collective Dynamics into Individual Utilities

We assert that there is a straight analogy between the traffic within a navigable resource – namely, agents entrances and exits – and the queueing phenomenon [30]. Queueing theory is sometimes used in pedestrians flow simulation, especially in evacuation simulation [14]. This theory provides pragmatic mathematical tools to describe the quality of the traffic when many client users want to access a service provider with limited capacity. It is possible to estimate the quality of the traffic through stochastic measures like, *mean service times*, *mean delays*, *mean number of users*, etc., if the probabilities distribution of departure and arrival times of users are known.

Here, we assimilate a navigation resource to a *provider*, agents to *users*, and discretized movements to *services* provided by resources. If an observer watches the entrance and exit times of transiting agents within navigation resources during the simulation, we can derive the following measures relatively to the observer by exploiting the queueing theory (the exponent "o" means that the estimation is relative to the observer):

$$D^o_{r_i,r_j} = q^o_{r_i} \cdot \left(\lambda^o_{r_i,r_j}\right)^{-1} \quad (5)$$

$$S^o_{r_i,r_j} = \left(\mu^o_{r_i,r_j}\right)^{-1} \quad (6)$$

$$v^o_{r_i,r_j} = L_{r_i,r_j} \cdot \left(S^o_{r_i,r_j}\right)^{-1} \quad (7)$$

- $\lambda^o_{r_i,r_j}$ and $\mu^o_{r_i,r_j}$ represent respectively arrival and departure frequencies of users travelling from $r_i$ to $r_j$
- $q^o_{r_i}$ is the mean number of users within $r_i$ – equation (5) is derived from Little's formula [30, p. 85]
- $L_{r_i,r_j}$ is the average length of $(r_i, r_j)$

An acceptable parallel would be therefore to associate an observer to each agent in order to derive the dynamic information variables relatively to agents. But for a real-time simulation perspective, this choice is risky in terms of memory use. This is why we introduce a single instance of a *traffic module*, that observes the arrival and departure of agents for each navigation resource, and compute dynamic information variables for them when requested. Agents interact with each others through the traffic module, by sending notifications when they enter or exit a navigation resource. Notifications are used by the traffic module to historize movements within each resource, and every agent can access dynamic information variables of resources that are within its *sensor range*. Note that it is possible to distribute several modules over the navigable space in order to process notifications efficiently.

Finally, we use (8) and (9) to consider agents' individual parameters in the explicit formulation of our dynamic information variables.

$$q_r = q^o_r \quad (8)$$

For any dynamic variable $M_{r_i,r_j}$:

$$M_{r_i,r_j} = M^\perp_{r_i,r_j} + \frac{q_{r_i} + q_{r_j}}{2 \cdot q_{max}} \cdot \left(M^o_{r_i,r_j} - M^\perp_{r_i,r_j}\right) \quad (9)$$

- $M^o_{r_i,r_j}$ is the value of the variable as computed by the traffic module
- $M^\perp_{r_i,r_j}$ is the value of the variable computed by the agent as if there was no traffic within $r_i$ and $r_j$ – i.e. by considering only its individual parameters and topological data. Note that if we assume $V_{pref}$ as the preferred velocity of an agent from $r_i$ to $r_j$, we can derive:

$$D^\perp_{r_i,r_j} = \frac{L_{r_i,r_j}}{V_{pref}} \quad (10)$$

$$S^\perp_{r_i,r_j} = D^\perp_{r_i,r_j} \quad (11)$$

$$v^\perp_{r_i,r_j} = V_{pref} \quad (12)$$

- $q_{max}$ is the maximum possible size of the resource (number of users)

Note that equation (9) formalizes two intuitive facts:

1. The more relevant the traffic, the more macroscopic is the measure of the variable
2. The less relevant the traffic the more individual is the measure.

In the next section, we present the specification of a generic multi-agent model that performs real-time simulations according to our formulation.

## 3.2 A Generic Multi-Agent Model

A key point for the specification of our model is the action theory to be used to implement the situated agents' behaviours. Our approach relies on the *influence reaction principle* proposed in [1], where there is a clear distinction between influences, which are produced by agents' behaviours, and the reaction of the environment. Precisely, our model specification is inspired from the IRM4S – Influence Reaction Model for Simulation [15] – which is a concretization of [1]'s theory for real-time simulation. In IRM4S, two distinct dynamics are coupled: *agents* generate influences to modify their representation in the environment, and *environment* reacts to all influences according to *natural laws*, and updates all the agents' representations. Here, we adapt the environment architecture to include a physics engine that updates agents' representations and a traffic module that mediate interaction between agents.

Figure 2 represents the global architecture of our framework.

The **physics engine** is responsible for the dynamics of agents' bodies. It updates bodies' positions and instantaneous speeds with respect to influences provided by agents, and accounts for shocks and collisions.

The **traffic module** is responsible for the dynamic information variables maintenance. It defines the topological structure that represents the continuous space, and mediate interactions between agents while they navigate.

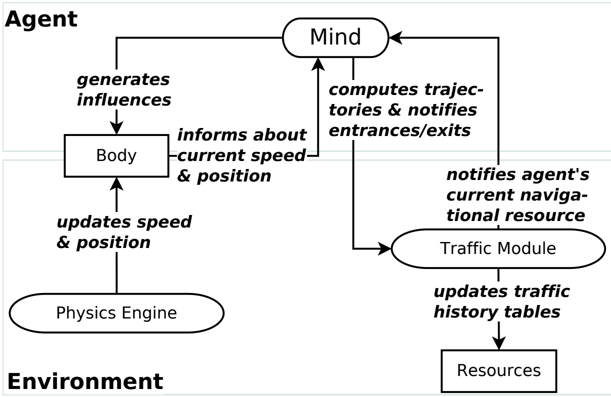
An **agent** is a relationship between a **mind** and a **body**. The mind dynamically maintains the energy most efficient path, relatively to the agent, and influences the body to follow the path until it reaches the destination.

We now detail the most important features of the model.

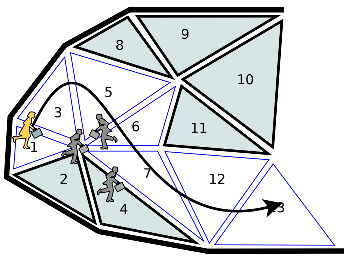
### 3.2.1 Agent

Our agent's model defines a *dynamic search algorithm* and an *influences set generation process* that guide the body towards the agent's destination.

The dynamic search algorithm starts from a current path and iteratively applies elementary moves, that consists in




(a) Architecture of our generic multi-agent model




(b) Presentation of the descriptive elements of an agent behaviour : the connected clear meshes represent the path of the agent. The bold curved arrow is the set of influences to be applied.

**Figure 2: Overview of our generic multi-agent model**

replacing links (discretized movements) in the current path, with alternative links in order to generate a more efficient path. Links replacement concerns only the resources that are sensed by the agent at the beginning of the search process. We experienced that such a dynamic search could be implemented efficiently by using an evolutionary search heuristic with a limited number of iterations [3]. Due to the lack of space, we do not detail the specification of this algorithm in this paper. We mostly focus on the architecture of the model.

An influences set could be visualized as a curved line that links the current agent's position with the farthest sensed resource on the path – see Figure 2(b). It formally represents the preferred velocities that the agent would take to reach its destination. The influences set computation could be handled by any linear interpolation algorithm.

To apply an influence to the body, the mind iteratively executes the algorithm 1. It selects the first influence from the current influences set and apply it to the body as the preferred velocity. When the body's position is updated by the physics engine, the mind notifies its movement to the traffic module which notifies back the travelled resources in order for the mind to update its path and, therefore, the current influences set.

### 3.2.2 Physics Engine

The physics engine updates bodies' positions and instantaneous speeds according to the velocity obstacle paradigm.

---

**Algorithm 1:** Application of influences

**Data**:
*pos*: mind's current assumed position ;
$I$: ordered set of influences ;
*path*: current path ;

**1** Select the first influence from $I$, $V_{pref}$ ;
**2** Set $V_{pref}$ to the body as the preferred velocity ;
**3** Get the new position $pos^{new}$ computed by the physics engine ;
**4** Notify the traffic module with $(pos, pos^{new})$ to get the travelled resources ;
**5** Update *path* according to the travelled resources ;
**6** Update $I$ ;
**7** Set $pos^{new}$ as the mind's assumed position ;

---

Given a preferred speed it computes the closest instantaneous velocity that allows a collision-free navigation in regards to all the dynamic obstacles.

### 3.2.3 Traffic Module

The traffic module updates *traffic history tables* of the resources according to a *history time step*. A *traffic history table* is a sliding window of predefined length that historizes agents' notifications. Two types of traffic history tables are associated to resources : *size* history table, to be used to estimate the mean number of users within the resource, and *transition* history tables, to be used to evaluate arrival or departure frequencies – a transition history table is associated to each discretized movement. When the traffic module is notified by an agent – with the mind's assumed position and the body's new position – it builds back the travelled resources chain to the agent, and stores them in a notification list. The notifications list is then processed at each history time step to maintain the traffic history tables of the travelled resources. We use a temporary classification for travelled resourced, labelled "*Active*", to process notifications efficiently. *Active* resources are resources that contains non zero values in their respective traffic history tables. At each history time step, only *Active* resources are maintained according to the algorithm 2.

## 4. EVALUATION

We have implemented our model in C++ on a standard MS machine – Intel E6550 dual core with a 2.33GHz processor and 2GB of memory – and carried out some experiments that highlight the most interesting features of our work, comparing to classical microscopic models. We have chosen the latest version of the RVO model, optimized for collision avoidance and CPU performances [25], to run series of comparative evaluations on selected benchmarks. The RVO model exploits the velocity obstacle paradigm as the underlying navigation principle, and performs within a multi-agent framework. We used the same type of collision avoidance algorithms to design a physics engine that matches our specification. The discretization of the space into triangle meshes has been realized with the freefem++ software [2]. Here, agents are physically represented as 2d disks of predefined radius and each resource cannot contain more than four agents.

[2]www.freefemplusplus.org

**Algorithm 2:** Traffic history table maintenance

```
Data:
Actives: "Active" resources list ;
Notifications : list of the travelled resources ;
foreach r ∈ Actives do
    Set the current history index value to 0 for every
    traffic history table of r ;
end
foreach c ∈ Notifications do
    Update Actives with the new travelled resources ;
    if |c| == 1 then  /* c has only one resource r_c */
        Q(r_c) ← size history table of r_c;
        increment the current history index value of
        Q(r_c) ;
    else  /* c has discretized movements (r_i, r_j) */
        foreach (r_i, r_j) ∈ c do
            T^{r_j}(r_i) ← transition history table of (r_i, r_j);
            Q(r_j) ← size history table of r_j;
            Q(r_i) ← size history table of r_i;
            decrement the current history index value of
            Q(r_i) ;
            increment the current history index value of
            T^{r_j}(r_i) ;
            increment the current history index value of
            Q(r_j) ;
        end
    end
end
Ignore non "Active" resources for the next step ;
```

We conducted two types of evaluations:

1. an online interview: we have invited volunteers to compare the performances of both model on low-density scenarios.

2. a validation of two well-known collective behaviours witnessed in highly congested crowds: the edge [23] and the fingering effects [28]

## 4.1 Online Interview

We have uploaded an online interview [3] to compare both models on several scenarios among the most frequently mentioned – see [12]. For each scenario, a pair of videos showing the performances of our model (labelled "GMAM") and RVO has been uploaded, and participants were invited to assign a comparative note among the following:

1. "none": **none** of the video is credible.

2. "++ credible": the left/right side video is **much more** credible

3. "+ credible": the left/right side video is **more** credible

4. "equally credible": both videos are **equally** credible

To ensure an objective comparison, the underlying model for each video has been hidden to participants, and videos were presented in a random order from one scenario to another. Figure 3 presents the results of the interview for the following benchmarks :

[3]www-desir.lip6.fr/˜simokanmeugne/evaluation0.html

1. *"Same Direction"*: a group of pedestrians walking in the same direction

2. *"Crossing"*: a crossing between two groups of pedestrians walking in opposite directions

3. *"Fast and Slow"*: a fast pedestrian walking behind a group of slow pedestrians

4. *"Narrow Passage"*: a group of pedestrians taking a narrow passage

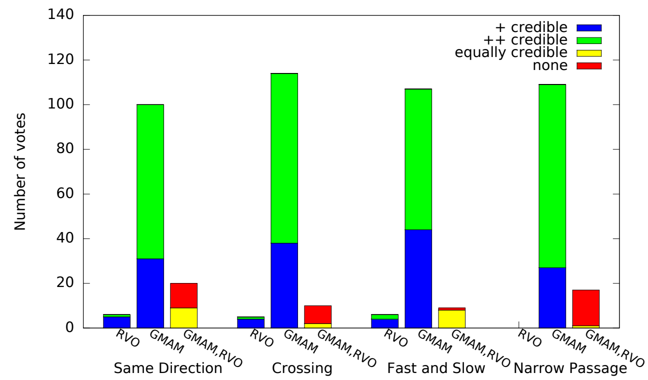



**Figure 3: Results of the online interview. Our model is labelled as "GMAM"**

A total of 140 participants completed the interview. Most of them (73%) were students or academics from our university. Results of the interview show that participants massively classified our model as the most credible for the given benchmarks. Hereinbelow, we justify the most relevant features of our model comparing to RVO.

1. *"Same Direction"*: our agents plan away from lateral and front resources for more efficiency. This results into emergent *V-like patterns* that we can witness in real life [16].

2. *"Crossing"*: less occupied and more fluid resources offer a better individual utility according to our formulation of the navigation task. As result, our agents prefer such resources in this benchmark and self-organize into *unidirectional lanes*.

3. *"Fast and Slow"*: the fastest agent, in our model, *overtakes* as soon as it gets close to the slow pedestrians group while the RVO agent passes in the middle of the group. Our agent plans for the surrounding resources, since they have better utility values relatively to its preferred speed.

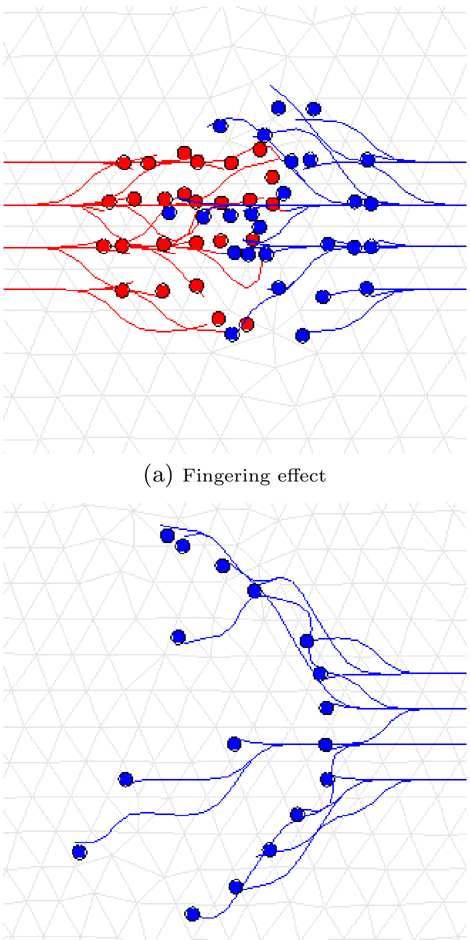
4. *"Narrow Passage"*: The more agents arrive at the entrance of the passage, the more the entrance's surrounding resources become congested. As result, our agents steer back to avoid congested areas at the entrance of the passage, while RVO agents spread laterally on the borders.

Next, we evaluated the performances of both models against two well-studied collective behaviours:

1. Edge effect: for unidirectional flows of pedestrians, *sides move faster than the center of the crowd* [23, 5, 20].
2. Fingering effect: for bidirectional flows, *pedestrians self-organize into unidirectional lanes to limit conflicts with the oncoming flow* [23, 28].

## 4.2 Collective phenomenon

Figures 4(a) and 4(b) illustrate how our model renders the fingering effect and the edge effect for low-density scenarios. The goal of this second evaluation is to generalizes the results for highly congested crowds.



(a) Fingering effect

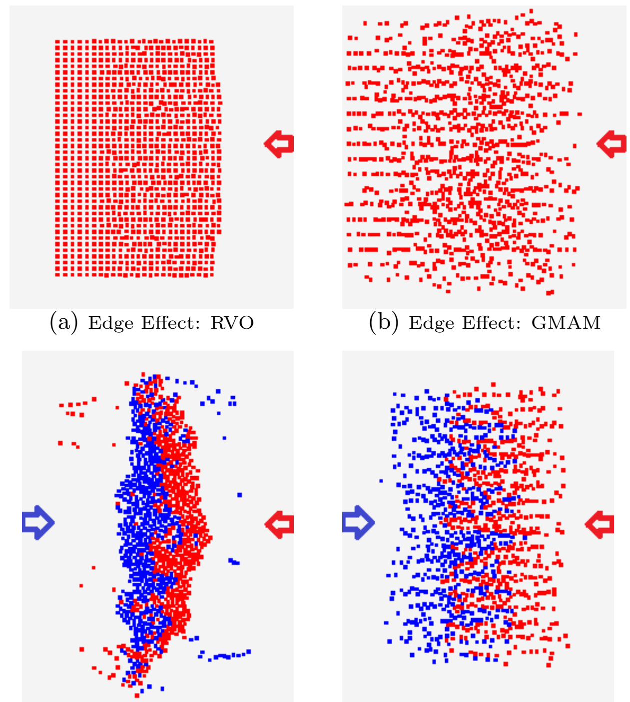
(b) Edge effect

**Figure 4: Illustration of our model performances against the edge effect (b) and the fingering effect (a) for low-density scenarios. Agents and their influences sets are coloured according to the direction of the movement. Here, the blue colour is for agents moving from the left to the right and the red colour, for agents moving from the right to the left.**

To reproduce highly congested crowds for this second evaluation, we realized four simulations of one thousand agents in restricted areas : for the edge effect, one thousand agents moving in the same direction, and for the fingering effect, a crossing between two groups of five hundred agents moving in two opposite directions. Figure 5 gives an overview of the differences between the performances of both models.

We can see that our model (Figures 5(b) and 5(d)) matches the descriptions of the collective behaviours better than RVO (Figures 5(a) and 5(c)).

Figure 5(d) illustrates self-organization into unidirectional lanes. Figure 5(b) shows side agents deviating from the center of the crowd and a more important concentration of agents in the middle of the crowd. These are encouraging results which prove that our model can produce credible results even for highly congested crowds.



(a) Edge Effect: RVO

(b) Edge Effect: GMAM

(c) Fingering Effect: RVO

(d) Fingering Effect: GMAM

**Figure 5: Validating the fingering effect and the edge effect**

# 5. CONCLUSION AND PERSPECTIVES

We proposed a generic multi-agent model for real-time simulation of a potentially highly congested crowd of autonomous pedestrians. We are interested in reproducing credible walking paths in real-time regardless of the number of agents. Our model originates from the principle of least effort applied to human walking behaviours and uses the influence and reaction principle to implement agents' behaviours. Agents communicate through a traffic module to dynamically maintains energy efficient paths, while being subject to a physics engine which updates their positions and instantaneous speeds.

The different experiments that we have made show encouraging results in terms of credibility. The dynamic planning algorithm that we used in combination with a traffic module give more insight to the agents and favours the emergence of complex individual and group behaviours like overtaking and V-like formations. Moreover, our model performs better than a classic microscopic model (RVO) when the number of agents increases, and reproduces some well-known collective behaviours like the fingering and the edge effect.

As short-term perspectives, we intend to work on the dynamics search calibrations in order to evaluate our work in terms of CPU performances. As mean-term perspectives, it could be interesting to study resource aggregation techniques to allow hierarchical planning. Deducing dynamic information for aggregated resources could be done the same way as for elementary resources, i.e. computed from agents' notifications. Also, we want to extend our formulation in order to account for time constraints and emergency situations. The concept of generalized cost developed in [8] provides interesting insights for that purpose. A long-term perspective is to work on the concept of *resource policy* to describe complex resource in terms of service quality. A resource policy could describe how a resource should be used. This could be helpful to elaborate richer urban simulations and integrate complex transports facilities like escalators, elevators, etc.

# 6. REFERENCES


[1] J. Ferber and J.-P. Müller. Influences and reaction: a model of situated multiagent systems. In *Proceedings of Second International Conference on Multi-Agent Systems (ICMAS-96)*, pages 72–79, 1996.

[2] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using the relative velocity paradigm. *[1993] Proceedings IEEE International Conference on Robotics and Automation*, 1993.

[3] F. Glover and M. Laguna. Tabu Search. *Journal of computational biology a journal of computational molecular cell biology*, 16(12):1689–703, 1997.

[4] E. Goffman. The Individual As A Unit. Relation in Public: Microstudies of the Public Order. *Allen Lane The Penguin Press*, 1972.

[5] S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. Lin, and D. Manocha. PLEdestrians : A Least-Effort Approach to Crowd Simulation. *Computing*, pages 119–128, 2010.

[6] D. Helbing and P. Molnar. Social Force Model for Pedestrian Dynamics. *Physical Review E*, 51(5):4282–4286, 1998.

[7] D. Helbing, P. Molnár, I. J. Farkas, and K. Bolay. Self-organizing pedestrian movement. *Environment and Planning B Planning and Design*, 28(3):361–383, 2001.

[8] S. P. Hoogendoorn and P. H. L. Bovy. Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B: Methodological*, 38(2):169–190, 2004.

[9] R. L. Hughes. A continuum theory for the flow of pedestrians, 2002.

[10] C. L. Hull. Principles of behavior: An introduction to behavior theory. 1943.

[11] M. Kapadia, S. Singh, W. Hewlett, and P. Faloutsos. Egocentric affordance fields in pedestrian steering. *Proceedings of the 2009 symposium on Interactive 3D graphics and games I3D 09*, 1(212):215, 2009.

[12] M. Kapadia, M. Wang, G. Reinman, and P. Faloutsos. Improved benchmarking for steering algorithms. In *Motion in Games*, pages 266–277. Springer, 2011.

[13] I. Karamouzas, R. Geraerts, and M. Overmars. Indicative routes for path planning and crowd simulation. *Proceedings of the 4th International Conference on Foundations of Digital Games FDG 09*, 45:113, 2009.

[14] G. G. Lovas. Modeling and simulation of pedestrian traffic flow. *Transportation Research Part B: Methodological*, 28(6):429–443, 1994.

[15] F. Michel. The IRM4S model: the influence/reaction principle for multiagent based simulation. In IFAAMAS, editor, *6th international joint conference on Autonomous agents and multiagent systems*, pages 903–905, Honolulu, Hawai'i, USA, 2007. ACM.

[16] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PloS one*, 5(4), 2010.

[17] K. M. Newell and W. A. Sparrow. Metabolic energy expenditure and the regulation of movement economy, 1998.

[18] J. Ondrej, J. Pettré, A.-H. Olivier, and S. Donikian. A synthetic-vision based steering approach for crowd simulation. *ACM Transactions on Graphics*, 29(4):1, 2010.

[19] C. Reynolds. Interaction with Groups of Autonomous Characters. *Group*, 21(4):449–460, 2000.

[20] P. Saboia and S. Goldenstein. Crowd simulation: applying mobile grids to the social force model. *The Visual Computer*, 28(10):1039–1048, 2012.

[21] A. Schadschneider, W. Klingsch, H. Kluepfel, T. Kretz, C. Rogsch, and A. Seyfried. Evacuation Dynamics: Empirical Results, Modeling and Applications. 2008.

[22] W. A. Sparrow. The efficiency of skilled performance. *Journal of motor behavior*, 15(3):237–261, 1983.

[23] G. K. Still. *Crowd dynamics*. PhD thesis, University of Warwick, 2000.

[24] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. *ACM Transactions on Graphics*, 25(3):1160, 2006.

[25] J. Van Den Berg, S. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. *Robotics Research*, pages 1–16, 2010.

[26] J. Van Den Berg and D. Manocha. Reciprocal Velocity Obstacles for real-time multi-agent navigation. *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935, 2008.

[27] M. W. Whittle. *An Introduction to Gait Analysis*. Butterworth-Heinemann, 2007.

[28] J. Zhang, W. Klingsch, A. Schadschneider, and A. Seyfried. Ordering in bidirectional pedestrian flows and its influence on the fundamental diagram. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(02):P02002, 2012.

[29] G. K. Zipf. *Human behavior and the principle of least effort: An introduction to human ecology*, volume 47. Addison-Wesley Press, 1949.

[30] M. Zukerman. Introduction to Queueing Theory and Stochastic Teletraffic Models. *Networks*, 2008:1–218, 2008.

# Influence of the interaction range on the stability of following models

Antoine Tordeux
a.tordeux@fz-juelich.de
JSC, FZ Jülich

Mohcine Chraibi
m.chraibi@fz-juelich.de
JSC, FZ Jülich

Armin Seyfried
a.seyfried@fz-juelich.de
JSC, FZ Jülich and Wuppertal University

## ABSTRACT

One proposes to analyze the stability of the uniform solutions of microscopic second order following models with $K \geq 1$ predecessors in interaction. We calculate general conditions for that the linear stability occurs, and explore the results with particular distance based pedestrian and car-following models. Non linear relations between $K$ and the stability are established.

## Categories and Subject Descriptors

G.1.7 [**Ordinary Differential Equations**]: Convergence and stability

## General Terms

Linear stability theory

## Keywords

Car-following model; Linear stability analysis of uniform solution; Number of predecessors in interaction

## 1. INTRODUCTION

Microscopic particles systems are frequently used to model pedestrian crowd or road traffic flow behaviors [3, 6]. Continuous models are defined with differential equations systems. The differential systems can be ordinary, stochastic or delayed, and of first or second order. The models have the uniform configuration (where the spacing and the speed are constant and equal) as equilibrium solution. The linear stability analysis of the uniform solutions allows to describe stationary state of the models [11]. The method consists in determining conditions on the parameters for which perturbations around the uniform solution vanish.

The number of predecessors in interaction is an essential parameter of the models. It is interpreted as an anticipation factor in traffic flow modeling [15]. Many car-following models with several predecessors in interaction exist in the literature [2, 10, 9, 12]. For pedestrian models, the parameter corresponds to the interaction range. In this paper, we calculate the linear stability for general ordinary models of second order with $K \geq 1$ predecessors in interaction. The results are explored with particular distance-based pedestrian and car-following models. They allow to justify when and why only a limited number of preceding agents needs to be taken into account when practically determining the acceleration of an agent.

### 1.1 Definition of the model

Let us consider an infinite 1D system of agents moving in the same direction. We denotes $n \in \mathbb{N}$ the index and $(x_n)$ the curvilinear positions of the agents. We suppose that the initial positions are such that the predecessor of the agent $n$ is the agent $n+1$.

The dynamics of the system are described by the second order model

$$\ddot{x}_n(t) = A\big(\dot{x}_n(t), x_{n+1}(t) - x_n(t), \dot{x}_{n+1}(t), \ldots, \\ x_{n+K}(t) - x_n(t), \dot{x}_{n+K}(t)\big). \quad (1)$$

The acceleration $A$ of the agent $n$ at time $t \geq 0$ depends on the speed, and on the speeds and distance spacings of the $K$ predecessors at the same time. We assume the function $A$ differentiable.

### 1.2 Uniform solution

For a given mean spacing $d > 0$, we suppose that a speed $v$ exists such that $A\,(v, d, v, 2d, v, \ldots, Kd, v) = 0$. Under this assumption, the uniform (or homogeneous) configurations $H$ such that for all $t \geq 0$ and all $n$

$$x^H_{n+1}(t) - x^H_n(t) = d, \qquad x^H_n(t) = x^H_n(0) + vt, \quad (2)$$

are solution of the system. It exists an infinity of uniform configurations, depending on the initial conditions. The linear stability of these solutions is investigated in this paper.

## 2. LINEAR STABILITY ANALYSIS

The literature distinguishes local stability analysis, for a finite line of agents with a leader traveling at a know speed, and global stability, for agents on a ring or on an infinite lane. The global stability conditions are more restrictive since they contain as well convective perturbations, that can locally vanish [14]. Here the global stability conditions are calculated on an infinite lane.

### 2.1 Characteristic equation

The stability conditions are calculated by studying the evolution of the differences $\tilde{x}_n(t) = x_n(t) - (x^H_n(0) + vt)$. An uniform solution $H$ is stable if $\lim_{t\to\infty} \tilde{x}_n(t) = \lim_{t\to\infty} \dot{\tilde{x}}_n(t) = 0$ for all $n$.

A first order Taylor approximation of (1) leads to the linear dynamics

$$\ddot{y}_n(t) = \sum_{k=1}^{K} \alpha_k (y_{n+k}(t) - y_n(t)) + \sum_{k=0}^{K} \beta_k \dot{y}_{n+k}(t). \quad (3)$$

where $\alpha_k = \frac{\partial A}{\partial d_k}(v, d, v, \ldots)$ and $\beta_k = \frac{\partial A}{\partial v_k}(v, d, v, \ldots)$.

A uniform configuration $H$ is linearly stable if $\lim_{t\to\infty} y_n(t) = \lim_{t\to\infty} \dot{y}_n(t) = 0$ for all $n$. If we solve (3) using the Ansatz $y_n(t) = \xi\, e^{\lambda t + in\theta}$, $\xi, \lambda \in \mathbb{C}^2$, $\theta \in \mathbb{R}$, we obtain the characteristic equation

$$\lambda^2 = \sum_{k=1}^{K} \alpha_k (e^{ik\theta} - 1) + \lambda \sum_{k=0}^{K} \beta_k e^{ik\theta}. \quad (4)$$

$H$ is linearly stable if the non nil roots of the characteristic equation have strictly negative real parts.

## 2.2 Linear stability condition

The characteristic equation is the complex polynomial equation with coefficients $(\nu_\theta, \mu_\theta, \sigma_\theta, \rho_\theta) \in \mathbb{R}^4$

$$\lambda^2 + w_\theta \lambda + z_\theta = 0, \quad w_\theta = \mu_\theta + i\sigma_\theta, \quad z_\theta = \nu_\theta + i\rho_\theta \quad (5)$$

with $\mu_\theta = -\sum_{k=0}^{K} \beta_k c_{k\theta}$, $\nu_\theta = \sum_{k=1}^{K} \alpha_k (1 - c_{k\theta})$, $\sigma_\theta = -\sum_{k=1}^{K} \beta_k s_{k\theta}$, $\rho_\theta = -\sum_{k=1}^{K} \alpha_k s_{k\theta}$, using the notations $c_x = \cos x$ and $s_x = \sin x$.

The sufficient and necessary conditions for that a polynomial with complex coefficients have all its zeros in the half-plane $\Re(\lambda) < 0$ are given in [4, Th. 3.2]. The results are a generalization of the so-called Hurwitz conditions for polynomials with real coefficients. They are here $\sum_{k=0}^{K} \beta_k < 0$ and

$$\mu_\theta > 0, \quad \mu_\theta(\nu_\theta \mu_\theta + \rho_\theta \sigma_\theta) - \rho_\theta^2 > 0, \quad \theta \in ]0, \pi]. \quad (6)$$

The condition is general and can be rediscovered in [13] with a model with one predecessor, or in [10] with the multi-anticipative optimal velocity model.

# 3. DISTANCE BASED MODELS

Many pedestrian dynamics models continuous in space are based on the superposition of a positive term to the desired speed and a negative repulsive one with the predecessors (see for instance [8, 5, 7])

$$\ddot{x}_n(t) = \frac{1}{\tau}(v_0 - \dot{x}_n(t)) - \sum_{k=1}^{K} f\left(x_{n+k}(t) - x_n(t)\right), \quad (7)$$

with $v_0, \tau > 0$ and $f$ a differentiable, positive, decreasing function on $\mathbb{R}^+$. Here, the repulsive force $f$ solely depends on the spacing.

With this model class, for a given mean spacing $d$, the equilibrium speed is $v = v_0 - \tau \sum_{k=1}^{K} f(kd)$. The speed $v$ depends on $v_0, K, d, \tau$ and $f(.)$ parameters. The first linear stability condition (6) is here $-1/\tau < 0$. It is always true and implies the preliminary assumption. The second condition (6) is

$$-\frac{1}{\tau^2} \sum_{k=1}^{K} f'(kd)\,(1 - c_{k\theta}) - \left( \sum_{k=1}^{K} f'(kd) s_{k\theta} \right)^2 > 0. \quad (8)$$

Note that $f'(d) \leq 0$ for all $d$ and thus the first term is positive and that the condition does not depend on $v$. The stability occurs for a relaxation time $\tau$ small enough. More precisely the homogeneous configurations are stable if and only if

$$0 < \tau < \tau_K = \inf_{\theta \in ]0,\pi]} \tau_K^{(\theta)},$$
$$\text{with} \quad \tau_K^{(\theta)} = \left( \frac{-\sum_{k=1}^{K} f'(kd)\,(1 - c_{k\theta})}{\left(\sum_{k=1}^{K} f'(kd) s_{k\theta}\right)^2} \right)^{1/2}. \quad (9)$$

We have $\lim_{x\to\infty} f(x) = f(y) - \int_y^\infty |f'(u)|\ \mathrm{d}u = 0$ for all $y > 0$. This implies $\int_y^\infty |f'(u)|\ \mathrm{d}u < \infty$ since for all $y > 0$, $f(y) < \infty$. Using the Cauchy criteria and changing the variable, one then obtains

$$\sum_{k=1}^{\infty} |f'(dk)| < \infty, \quad d > 0. \quad (10)$$

This proves the absolute convergence of $\tau_K^{(\theta)}$ and $\tau_K$, and means that the stability condition at the limit $K \to \infty$ may be approximated for an finite value of $K$. It exist with this model class an intrinsic interaction range. The value of the range depends on the convergence speed of the series $\sum_k |f'(dk)|$. This point will be further investigated using well-know repulsive forces $f$.

## 3.1 Exponential and inverse models

Let firstly consider the exponential repulsive force with parameters $A, B > 0$ into (7)

$$f(d) = Ae^{-d/B}, \qquad f'(d) = -A/Be^{-d/B}. \quad (11)$$

This force is used in the social force model [8]. Because of the use of exponential decreasing, the interaction model is short range. We use the uni-dimensional parameter $u = d/B > 0$ and critical relaxation time

$$\tilde{\tau}_K^{(\theta)} = \sqrt{\frac{A}{B}}\tau_K^{(\theta)}. \quad (12)$$

We have with the exponential repulsive force (11) using (9)

$$\tilde{\tau}_K^{(\theta)}(u) = \left( \frac{\sum_{k=1}^{K} e^{-ku}\,(1 - c_{k\theta})}{\left(\sum_{k=1}^{K} e^{-ku} s_{k\theta}\right)^2} \right)^{1/2}. \quad (13)$$

$u$ is a shape parameter, while $A$ and $B$ are scale parameters for the stability.

The inverse repulsive force with parameters $A, B, q > 0$ is

$$f(d) = \frac{A}{(d/B)^q}, \qquad f'(d) = -\frac{qA/B}{(d/B)^{q+1}}. \quad (14)$$

This model is used in [5] with $q = 1$ and in [7] with $q = 2$. Here, the model can be short or long range depending on the value of $q$. It induces a polynomial convergence speed of $f'$ to zero, slower than the exponential speed of the model (11). This suggests higher number of pedestrians in interaction $K$ to stabilize the critical time $\tau_K$. We have with this model the dimensionless critical relaxation time

$$\tilde{\tau}_K^{(\theta)}(u, q) = \left( \frac{u^{q+1}}{q} \frac{\sum_{k=1}^{K}(1 - c_{k\theta})k^{-(q+1)}}{\left(\sum_{k=1}^{K} s_{k\theta} k^{-(q+1)}\right)^2} \right)^{1/2}. \quad (15)$$

Here again, only $q$ is a shape parameter.

## 3.2 Stability condition

The uniform solution (2) is linearly stable for the distance based models (11) and (14) if the relaxation time $\tau$ is strictly less than critical time $\tau_K = \inf_\theta \tau_K(\theta)$ (or if $\sqrt{A/B}\tau < \tilde{\tau}_K$). Thus we have to calculate the minimum of the functions $\theta \mapsto \tilde{\tau}_K(\theta)$ to determine the stability condition. Yet, the signs of the derivative of these functions are hardly analytically extracted. We investigate it numerically.

The critical time (13) of the exponential model (11) is plotted as a function of $\theta$ in figure 1. Here, $K$ varies from 1 to 25, and $u = 0.4$, 1 and 2.5. The $\tilde{\tau}_K^{(\theta)}$ are minimal at the limit $\theta \to 0$ for all $K$, *i.e.* $\tilde{\tau}_K = \lim_{\theta\to 0} \tilde{\tau}_K^{(\theta)}$ with the exponential force (11). Further numerical investigations (not shown here) confirm this observation.

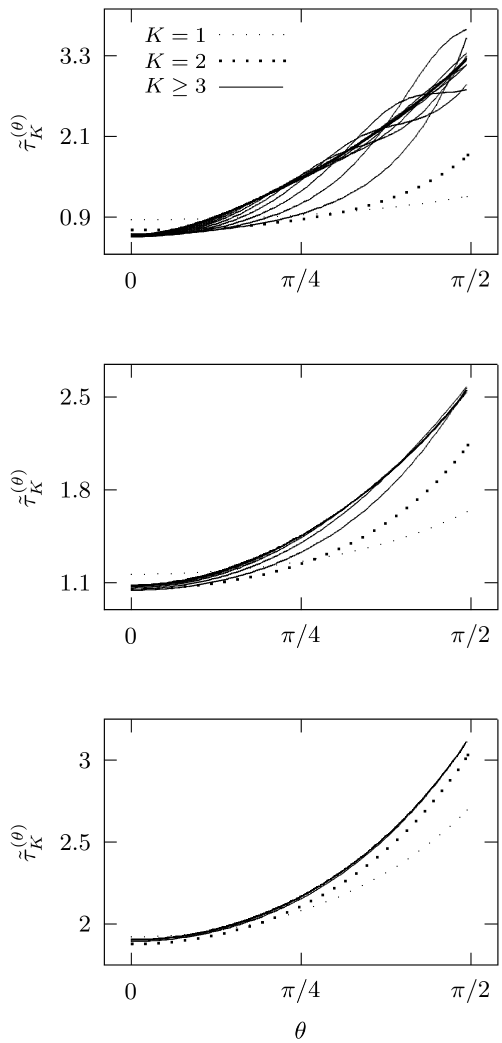



**Figure 1: The $\tilde{\tau}_K^{(\theta)}$, $K = 1, \dots, 25$, as a function of $\theta$ for the exponential model (11). From top to bottom $u = 0.4, 1, 2$.**

The critical time (13) for the inverse model (14) is plotted as a function of $\theta$ in figure 2 with $q = 1$, 2 and 3. The $\tau_K^{(\theta)}$ are minimal for $\theta \to 0$ when $K$ is low. For high values of $K$, the minimums of $\tau_K^{(\theta)}$ are reached for $\theta = \theta_{q,K} > 0$. Further results show that $\theta_{q,K}$ converge when $K$ increases. Therefore the wave's lengths the more unstable have characteristic values with the inverse model, if $K$ is large enough.

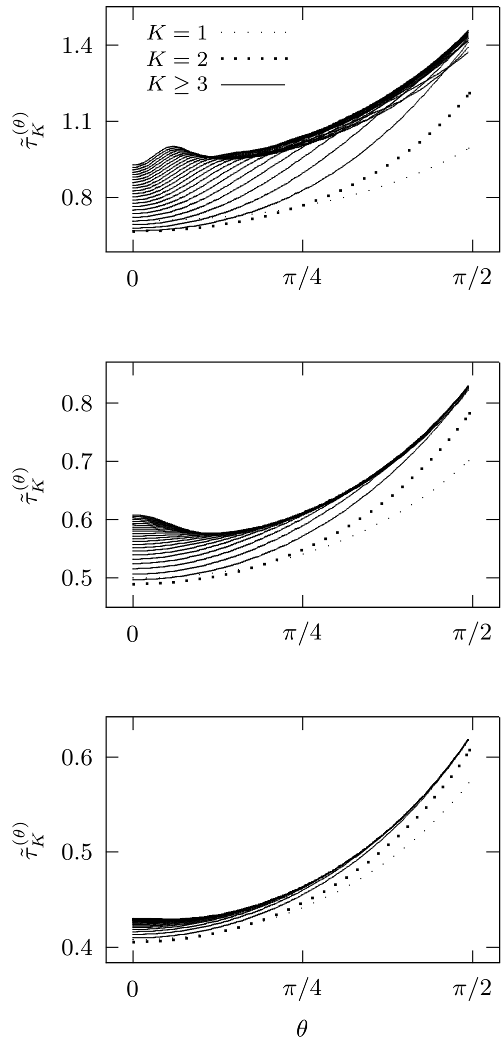



**Figure 2: The $\tilde{\tau}_K^{(\theta)}$, $K = 1, \dots, 25$, as a function of $\theta$ for the inverse model (14). From top to bottom $q = 1, 2, 3$; $u = 1$.**

## 3.3 Stability function of $K$

The dimensionless critical time $\tilde{\tau}_K$ delimits the border of the linear stability of uniform solutions. The stability occurs if $\sqrt{A/B}\tau$ is strictly smaller than $\tilde{\tau}_K$ (see (9)).

The dimensionless function $K \mapsto \tilde{\tau}_K$ is plotted in figure 3 for the model (11), and in figure 4 for (14). One can observe for both models that the critical time $\tau_K$ converges to a constant value through a single damped oscillation. This non linear relation between $K$ and the stability is surprising. Increasing the number of pedestrians in interaction firstly results as a decreasing of the stability (at least until $K = 2$). Then increasing $K$ increase $\tau_K$ and so the stability. The convergence of $\tau_K$ is relatively smooth with the model (11).

One observes a brusque transition with the model (14), when the minimum is reached for the $\theta_{q,K}$. The form and speed of the damping of function $K \mapsto \tilde{\tau}_K$ depend on parameter $u$ for the model (11), and on $q$ for (14).

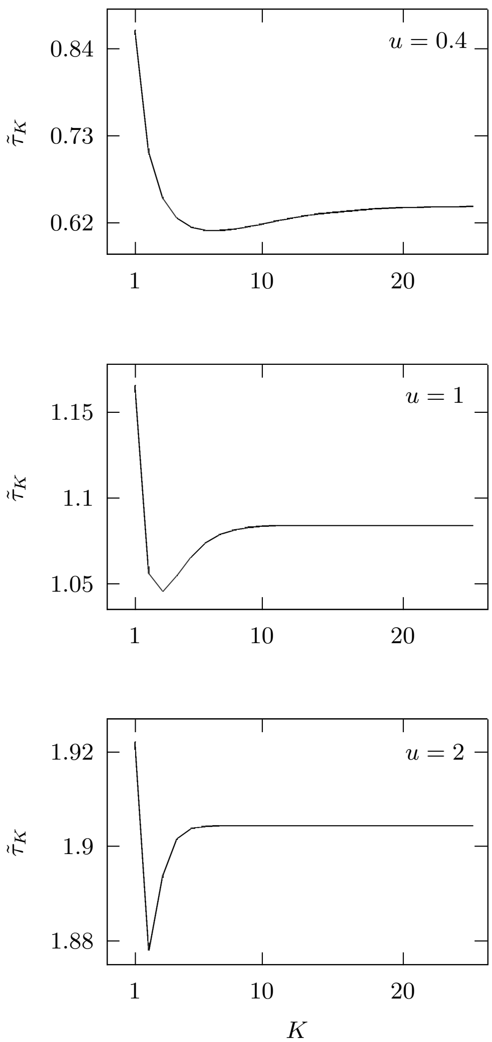



**Figure 3: $\tilde{\tau}_K = \inf_\theta \tilde{\tau}_K^{(\theta)}$ as a function of $K$ for the exponential-distance model (11) with $u = 0.4,\ 1,\ 2$.**

## 3.4 Proportion of variation

The figures 3 and 4 show damping oscillations of $\tilde{\tau}_K$ as $K$ increases. Here, we investigate the amplitude of the oscillation. For that purpose, we introduce the proportion

$$\varphi = 1 - \frac{\min_K \tilde{\tau}_K}{\max_K \tilde{\tau}_K} = 1 - \frac{\min_K \tau_K}{\max_K \tau_K}, \qquad (16)$$

that is the same for the initial and dimensionless critical time $\tau_K$ and $\tilde{\tau}_K$.

The proportion of variation $\varphi \in [0, 1]$ describes how the models depend on the number $K$. For $\varphi \approx 0$, the model poorly depends on the interaction range, i.e. the stability condition as $K \to \infty$ is well approximated using few predecessors ($K$ small), and oppositely for $\varphi \approx 1$. For both

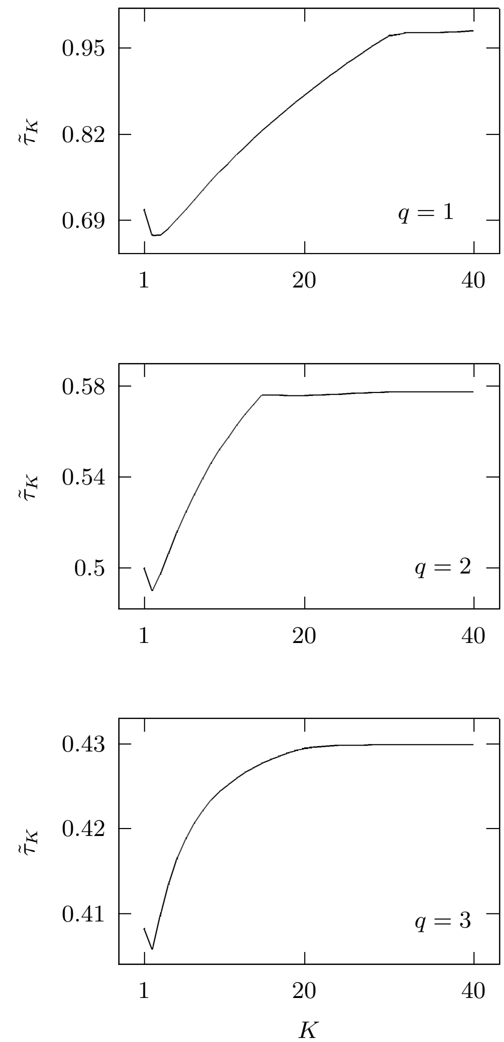



**Figure 4: $\tilde{\tau}_K = \inf_\theta \tilde{\tau}_K^{(\theta)}$ as a function of $K$ for the inverse model (14) with $q = 1,\ 2,\ 3$; $u = 1$.**

models (11) and (14), $\varphi$ does not depend on $A$ and $B$. It depends on $u$ for the exponential model (11), while it depends on $q$ for the inverse model (14), but not on $u$.

In figure 5, the proportion of variation $\varphi$ is plotted as a function of $u$ for the model (11) (top plot), and as a function of $q$ for the model (14) (bottom plot). $\varphi$ tends to zero as the dimensionless mean spacing $u$ increases within model (11). This means that for low density level, the stability condition at the limit $K \to \infty$ can be well estimated using few predecessors. For high densities, the variability of $\tilde{\tau}_K$ is more important. The same phenomena occurs as $q$ increases within model (14). For short range model where $q$ is high, few predecessor in interaction are sufficient to estimate the stability condition as $K \to \infty$ and oppositely. Surprisingly, the proportion of variation does not depend on the density level with the inverse model. This changes in the case when the distance spacing $d$ is taken as $d - \ell$ to take into account the size $\ell > 0$ of the pedestrians.

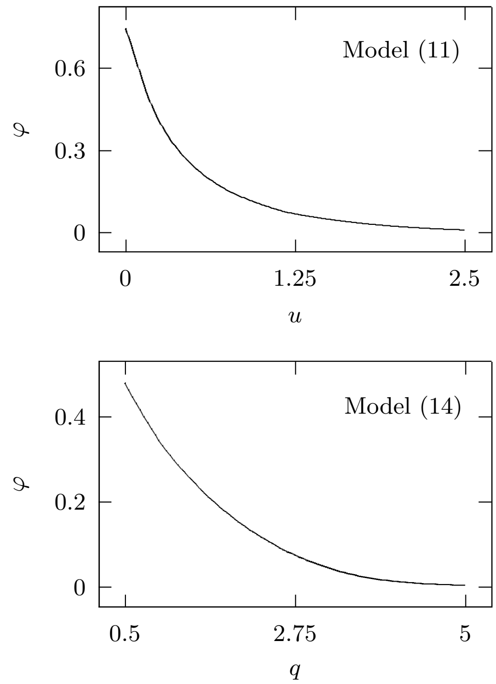



**Figure 5: Proportion $\varphi$ of variation of $\tau_K$ as $K$ increases. Left, function of $u$, model (11). Right, function of $q$, model (14).**

## 3.5 Stability function of the parameters

The function $\tilde{\tau}_K$ depends on the parameter $u$ for the model (11), and on $(u, q)$ for (14). For the model (11), the function $u \mapsto \tilde{\tau}_K(u)$ increases to infinity as $u$ increases. This means that the stability increase as the distance spacing increases, for any $K$. One has explicitly

$$\frac{\partial \tilde{\tau}_K^{(\theta)}}{\partial u}(u) = \frac{\tilde{\tau}_K^{(\theta)}(u)}{2} g_K^{(\theta)}(u) > 0 \tag{17}$$

with $g_K^{(\theta)}(u) = -\frac{\sum_{k=1}^K k\, e^{-uk}(1-c_{k\theta})}{\sum_{k=1}^K e^{-uk}(1-c_{k\theta})} + \frac{2\sum_{k=1}^K k\, e^{-uk} s_{k\theta}}{\sum_{k=1}^K e^{-uk} s_{k\theta}}$.

We have $g_1^{(\theta)}(u) = 1$ for all $u$, while $\lim_u g_K^{(\theta)}(u) = 1$ for all $u$ and all $K > 1$.

In top figure 6, the increasing critical time $\tilde{\tau}_K(u)$ at the limit $K \to \infty$ are compare to the time $\tilde{\tau}_1(u)$ for $K = 1$. One has $\lim_K \tilde{\tau}_K(u) < \tilde{\tau}_1(u)$ for all $u$, while, as expected since the proportion of variation tends to zero, $\lim_u \tilde{\tau}_1(u) = \lim_u \tilde{\tau}_K(u)$ for any $K$.

For the model (14), the function $u \mapsto \tilde{\tau}_K^{(\theta)}(u, q)$ also increases as $u$ increases since

$$\frac{\partial \tilde{\tau}_K^{(\theta)}}{\partial u}(u, q) = \frac{q+1}{u} \tilde{\tau}_K^{(\theta)}(u, q) > 0. \tag{18}$$

The relation $q \mapsto \tilde{\tau}_K^{(\theta)}(u, q)$ is more complicated. For $u < 1$, the function decreases to zero as $q$ increases. This means that stability never holds for any $\tau$ for enough high $q$. For $u = 1$, $\tilde{\tau}_K^{(\theta)}(1, q)$ tends to a constant value, while, for $u > 1$, the relation, successively decreasing and increasing, admits a minimum for certain $q$ depending on $u$. One has

$$\frac{\partial \tilde{\tau}_K^{(\theta)}}{\partial q}(u, q) = \frac{\tilde{\tau}_K^{(\theta)}(u, q)}{2} \left( \ln u - \frac{1}{q} + h_K^{(\theta)}(q) \right), \tag{19}$$

$h_K^{(\theta)}(q) = -\frac{\sum_{k=1}^K \ln k\, k^{-(q+1)}(1-c_{k\theta})}{\sum_{k=1}^K k^{-(q+1)}(1-c_{k\theta})} + \frac{2\sum_{k=1}^K \ln k\, k^{-(q+1)} s_{k\theta}}{\sum_{k=1}^K k^{-(q+1)} s_{k\theta}}$.

For $K = 1$, $h_1^{(\theta)}(q) = 0$ for all $q$, and the sign of $\partial \tilde{\tau}_1^{(\theta)}/\partial q > 0$ is the sign of $\ln u - 1/q$. It is negative for all $q$ if $u < 1$. If $u > 1$, $\partial \tilde{\tau}_1^{(\theta)}/\partial q < 0$ for $q < 1/\ln u$, and $\partial \tilde{\tau}_1^{(\theta)}/\partial q > 0$ for $q > 1/\ln u$. The $\tilde{\tau}_1^{(\theta)}(u, q)$ are minimum for $q = 1/\ln u$. Comparable properties are obtained for $K > 1$. One has $-1/q + h_K^{(\theta)}(q) \to -\infty$ as $q \to 0$, $\partial \tilde{\tau}_K^{(\theta)}/\partial q$ is firstly negative, and it is of the sign of $\ln u$ as $q$ increases since $-1/q + h_K^{(\theta)}(q) \to 0$ as $q \to \infty$.

The critical time $\tilde{\tau}_K(u, q)$ at the limit $K \to \infty$ is compared to $\tilde{\tau}_1(u, q)$, for $u = 0.4$ and $u = 2$, and as a function of $q$ in bottom figure (6). Here $\tilde{\tau}_1(u, q) < \lim_K \tilde{\tau}_K(u, q)$ for all $u, q$, while, as expected, $\lim_q \tilde{\tau}_1(u, q) = \lim_q \tilde{\tau}_K(u, q)$ for any $K$ and $u$.

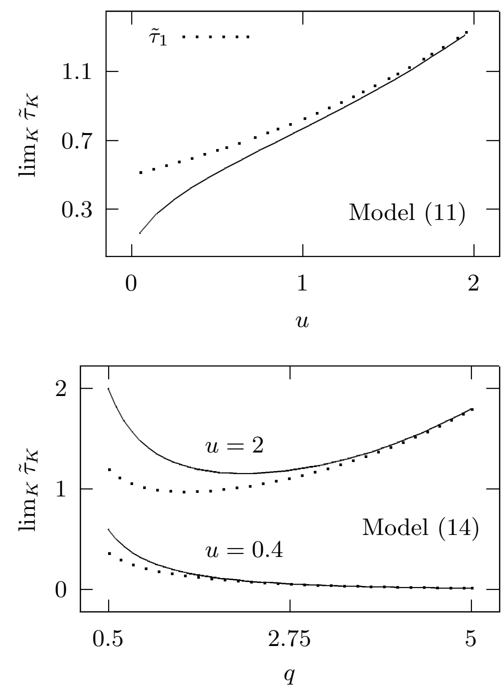



**Figure 6: Dimensionless critical relaxation time $\tilde{\tau}_K = \inf_\theta \tilde{\tau}_K^{(\theta)}$ for $K = 1$ (dotted lines), and at the limit $K \to \infty$ (continuous lines). Top, function of $u$, model (11). Bottom, function of $q$, model (14).**

## 3.6 Optimal velocity model

The multi-anticipative optimal velocity model with $K \geq 1$ predecessors in interaction [10] is

$$\ddot{x}_n(t) = \sum_{k=1}^{K} a_k \left\{ V\left(\frac{1}{k}(x_{n+k}(t) - x_n(t))\right) - \dot{x}_n(t) \right\}. \tag{20}$$

Here $(a_k) \in \mathbb{R}_+^K$. With this model, the equilibrium speed $v$ corresponding to the mean spacing $d$ is $v = V(d)$. The first condition (6) is $\sum_{k=1}^{K} a_k > 0$. It is always true and implies the preliminary assumption. The second condition (6) is, after rearranging

$$0 < V' < \frac{\left(\sum_{k=1}^{K} a_k\right)^2 \sum_{k=1}^{K} \frac{a_k}{k}(1 - c_{k\theta})}{\left(\sum_{k=1}^{K} \frac{a_k}{k} s_{k\theta}\right)^2} \qquad (21)$$

(see [10, Eq. (14)]).

Note that the case $K = 1$ corresponds to the well know Optimal Velocity model [1]. For this model, the condition is $V'(d) < a_1/(1 + c_\theta)$. Since $1/(1 + c_\theta) > 1/2$, the condition holds for all $\theta \in ]0, 2\pi[$ if $V'(d) < a_1/2$ (see [1]).

If we assume that $1/a_k = \tau k^q$ with $\tau > 0$ and $q \geq 0$ a parameter calibrating the interaction range (in a similar way than with the inverse model (14)), one obtains the condition

$$0 < \tau V' < \frac{\left(\sum_{k=1}^{K} k^{-q}\right)^2 \sum_{k=1}^{K} (1 - c_{k\theta})\, k^{-(q+1)}}{\left(\sum_{k=1}^{K} s_{k\theta} k^{-(q+1)}\right)^2} =: \tilde{\tau}_K^{(\theta)}. \qquad (22)$$

The mean spacing has only a role through the derivative of the optimal speed function that is a scale parameter. Only $q$ is a shape parameter. The expression of $\tilde{\tau}_K^{(\theta)}$ is comparable to the one of the inverse model (14), see (15). Here, the time is proportional to the square of $\sum_k k^{-q}$ and converges if and only if $q > 1$. In this case, the forms of the functions $\theta \mapsto \tilde{\tau}_K^{(\theta)}$ are comparable to the ones obtained with the inverse model (14) (see in figure 2). The functions $K \mapsto \tilde{\tau}_K = \inf_\theta \tilde{\tau}_K^{(\theta)}$ are also comparable with the difference that the functions are always increasing, with no damped oscillation. With the OV model (20), increasing the number of predecessors in interaction results in an increase of the stability, for any $V'(d) > 0$.

The proportion of variation of the critical time $\tilde{\tau}_K$ as $K$ varies, denoted $\varphi$, is not defined when $q \leq 1$ since the $\tilde{\tau}_K$ diverges (it could be equal to 1). For $q > 1$, the proportion tends to zero as $q$ increases. As expected, and as the inverse model (14), see in top figure 5, the influence of $K$ decreases as the model becomes short range (i.e. as $q$ increases). The critical time does not depends on $q$ for $K = 1$. For any $K > 1$, the times decreases as the $q$ increases. The stability is negatively influenced by the range $q$. The constant minimal value for $K = 1$ corresponds here to the limit as $q$ increases of the critical time $\tilde{\tau}_K$ for all $K > 1$ ($\lim_q \tilde{\tau}_K(q) = \tilde{\tau}_1 > 0$, see in bottom figure 7). This means that, oppositely to the inverse model (14) with $u < 1$, the OV model can remain stable at the limit $q \to \infty$, for any $K$.

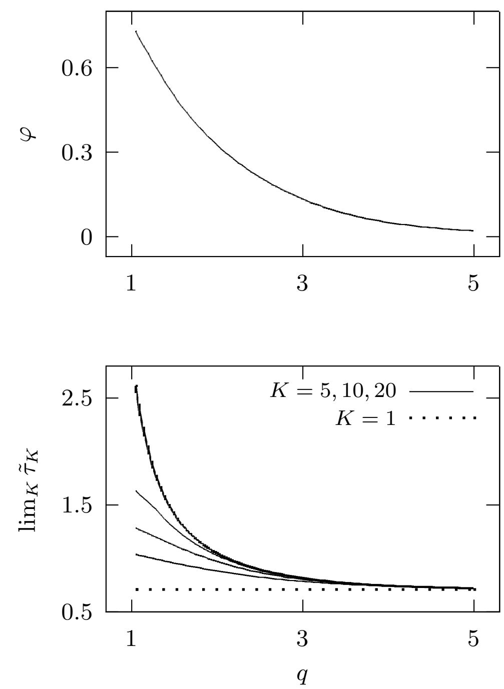



**Figure 7: Top, the proportion of variation of the critical time with the OV model (20). Bottom dimensionless critical time for $K = 1, 5, 10, 20$ and at the limit $K \to \infty$ (thick line).**

## 4. SUMMARY AND CONCLUSION

Linear stability conditions of uniform solutions are calculated for a second order pursuit model, with $K \geq 1$ predecessor in interaction. The framework is general and includes many models used in pedestrian dynamics as well as in road traffic flow. The conditions are explored using particular pedestrian models for which the dynamics are the sum of an acceleration term to the desired speed, and a repulsive one with the predecessors, or with the well-known optimal velocity car-following model. For the pedestrian models, the acceleration to the desired speed is calculated using a relaxation process, while the repulsion is a sum over the predecessors in the interaction. The desired speed is a function of the spacing with the optimal velocity model. Within the different forms tested, the stability occurs for a small enough relaxation time $\tau$, smaller than a critical time $\tau_K$.

When the repulsive force depends solely on the distance spacing, the critical time converges as $K$ increases, with a damped oscillation. The role of the parameter $K$ on the stability threshold is not negligible when the repulsive term does not decrease sufficiently fast as the distance spacing $d$ increases (i.e. force $f(d) \propto e^{cd}$ or $1/d^q$ with low $c, q$). On the opposite, the proportion of variation of $\tau_K$ as $K$ varies is low when the interaction range model are short (i.e. high $c$ or $q$ parameters). Comparable properties are obtained within the car-following optimal velocity model if $q > 1$.

The number of predecessors in interaction in the pursuit modeling modify the stability conditions. As expected, the influence of the parameter depends on the form of the model. It exists finite interaction thresholds for the stability within distance based models. In a separate paper we will show that it is generally not the case when the models also depends on the speeds. The overview developed here could be useful regarding jam waves formation, for analysis or validation of pedestrian as well as car-following models.

## 5. REFERENCES

[1] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama. Dynamical model of traffic congestion and numerical simulation. *Phys. Rev. E*, 51(2):1035–1042, 1995.

[2] S. Bexelius. A extended model for car-following. *Transp. Res.*, 2(1):13–21, 1968.

[3] D. Chowdhury, L. Santen, and A. Schadschneider. Statistical physics of vehicular traffic and some related systems. *Phys. Rep.*, 329(4-6):199–329, 2000.

[4] E. Frank. On the zeros of polynomials with complex coefficients. *Bull. Amer. Math. Soc.*, 52(2):144–157, 1946.

[5] R.Y. Guo, S.C. Wong, H.-J. Huang, and W.H.K. Lam. A micoscopic pedestrian-simulation model and its application to intersecting flows. *Physica A*, 389(3):515–526, 2010.

[6] D Helbing. Traffic and related self-driven many-particle systems. *Phys. Mod. Phys.*, 73(4):1067–1141, 2001.

[7] D. Helbing, I.J. Farkas, and T. Vicsek. Freezing by heating in a driven mesoscopic system. *Phys. Rev. Lett.*, 84(6):1240–1243, 2000.

[8] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51(5):4282–4286, 1995.

[9] S. Hoogendoorn, S. Ossen, and M. Schreuder. Empirics of multianticipative car-following behavior. In *Transportation Research Record*, number 1965, pages 112–120, 2006.

[10] H. Lenz, C.K. Wagner, and R. Sollacher. Multi-anticipative car-following model. *Eur. Phys. J. B*, 7(2):331–335, 1999.

[11] G. Orosz, R.E. Wilson, and G. Stepan. Traffic jams : dynamics and control. *Proc. R. Soc. A*, 368(1957):4455–4479, 2010.

[12] A. Tordeux, S. Lassarre, and M. Roussignol. An adaptive time gap car-following model. *Transp. Res. B*, 44(8-9):1115–1131, 2010.

[13] A. Tordeux, M. Roussignol, and S. Lassarre. Linear stability analysis of fisrt-order delayed car-following models on a ring. *Phys. Rev. E*, 86(3):036207, 2012.

[14] M. Treiber and A. Kesting. *Traffic Flow Dynamics.* Springer, Berlin, 2013.

[15] M. Treiber, A. Kesting, and D. Helbing. Delays, inaccuracies and anticipation in microscopic traffic models. *Physica A*, 360(1):71–88, 2006.