

MULTI-AGENT COORDINATION AND ANTICIPATION MODEL TO DESIGN A ROAD TRAFFIC SIMULATION TOOL

Arnaud Doniec ^{ab} Stéphane Espié ^b René Mandiau ^a Sylvain Piechowiak

^a *LAMIH, University of Valenciennes, Le Mont-Houy, 59313 Valenciennes Cedex, France*

^b *INRETS, 2 avenue du Général Malleret-Joinville, 94114 Arcueil Cedex, France*

Abstract

The use of a multi-agent approach in the design of a traffic simulation tool is innovative since most of actual tools are still based on mathematical approach. In this paper we present a multi-agent approach to simulate in a realistic way the traffic phenomenon at junction. We use a multi-agent coordination approach which is improved by an anticipation algorithm based on constraints processing. The resulting tool is validated by comparison between simulated flow and real flow measured at a real junction.

1 Introduction

The use of a multi-agent approach allows to build complex softwares in various domains. Simulation tools are a perfect illustration of such a software. Applying the multi-agent paradigm helps to simulate complex and dynamic phenomena which are hardly expressible with a classical formalism like differential equations for instance.

In this paper, we focus on the traffic simulation issue which consists of reproducing in a realistic way the traffic phenomenon currently observed in reality. More precisely, we are interested in the development of microscopic traffic simulation tool which is able to reproduce the moves of a set of vehicles over a network with a limited capacity and a road infrastructure (traffic sign, road mark, etc). In this context, one of the hardest problems is the road intersection management.

This article deals with a multi-agent model which is able to simulate in a realistic way the crossing of an intersection by a set of vehicles. The first section gives a brief review of current traffic simulation tools and presents ARCHISIM: the traffic simulation tool developed at INRETS (French National Institute for Transport And Safety Research). The second section tackles the coordination techniques that have been integrated in ARCHISIM in order to simulate traffic in intersection. In particular, we outline an anticipation framework based on constraints processing and used to improve the multi-agent coordination. The last section deals with its validation. This one is based on a comparison between actual and simulated flows on a real crossroad.

2 A review of existing traffic simulation tools

Most of traffic simulation tools are based on a mathematical approach which consists of reproducing vehicles streams from car-following laws. These laws are differential equations, obtained empirically by regression using data collected at currently operating road section [15].

For the specific case of road junction, these tools simplify in a drastic way what happens inside the intersection. The usual approach of these tools uses a centralized scheduler running over the traffic flow of each arm of the crossroad. This scheduling process lets enter into the intersection only the vehicles whose trajectories are not in conflict. The simulated vehicles which reach the crossroad are stocked in a queue (one per axis) and the scheduler searches a gap to insert the head of each queue. This is done with respects of two constraints:

- the time between two consecutive vehicles must be sufficient to allow the crossing of a conflicting vehicles stream

- the different moves (inside the intersection) have to be limited in order to reduce the number of conflicts

Moreover each tool uses its own internal rules to limit the occupancy of the inner space of the intersection. In AIMSUN [1] for example, users can configure a specific parameter called "yellow box" and thus define a minimal speed that the vehicles inside the crossroad have to practice so that other vehicles can enter.

Such tools are sometimes sufficient for superficial traffic studies, but are not acceptable when the question is to mimic actual behavior. Indeed, no more than two vehicles are present inside the intersection at the same time which is not realistic at all. Moreover these tools allow neither a good accuracy of traffic flow (and jam...) for high density traffic, nor the simulation of a surrounding traffic for a driving simulator.

In opposition to this "mathematical approach", the "behavioral approach" produces an emerging traffic by managing interactions between various actors of the road situations (car drivers, pedestrians, road operator, ...). The resulting traffic is therefore the sum of all actors' individual actions. ARCHISIM is a behavioral traffic simulation tool developed by INRETS [9].

The computing model of ARCHISIM follows a multi-agent architecture. Each simulated driver is an autonomous software agent which evolves in a virtual environment and interacts with the other agents of the simulation performing its goals according to its skills and the current situation.

At each simulation step, an agent receives a set of information describing the surrounding situation in the environment. Thanks to these informations, each agent takes its own decision which results in a longitudinal and lateral acceleration for the next time step.

The novelty of ARCHISIM lies in the ability for a driving simulator to take part in the traffic simulation. In other words, ARCHISIM is able to generate a surrounding realistic traffic for a human driver in a simulator. The agents interact with the driving simulator like with any other vehicles of the simulation.

The traffic model of ARCHISIM has been validated for highway situation [8], [4]. Then, efforts have been undertaken to extend the traffic model of ARCHISIM into intersections management. A specific coordination mechanism has been proposed in [3] and we improve this mechanism by adding an anticipation mechanism. We describe this work in the next sections.

3 Multi-agent coordination applied to traffic simulation

3.1 An overview of coordination in MAS

When crossing an intersection a real driver has to solve its conflicts with other vehicles. In our simulation context, this driving task can be expressed as a multi-agent coordination issue: all agents reaching a crossroad have to coordinate their actions in order to avoid accidents and deadlocks.

Coordination concerns all local processes that allow to achieve a global state of the system. Many definitions exist in literature, each of them highlights a particular point of view. For Jennings, coordination is the process which allows an agent to reason about its own actions and the actions of other agents in order to ensure that the global system correctly behaves [11]. In this manner, coordination can be used to satisfy and guarantee some criteria of consistency without any global process. Malone defines coordination as all extra activities which are required to have interactions between agents [17].

Originally, first works on multi-agent coordination have investigated the coordination from a cooperative viewpoint. They made the assumption that all agents were motivated to achieve a common goal. We can quote for instance Lesser and Corkill's works whose model was based on exchanged messages describing aims, priorities and preference of the agents [14]. The coordination has also been expressed as a planning problem [10]. Each agent can formalize in a plan its actions and interactions in the future. In this context, the coordination enables the laying down of a multi-agent plan describing the global behavior of the system, relying on all individual plans. The coordination mechanism has therefore to identify and find: the possible link between actions, the sequential and parallel executions of actions, the potential conflicts which can occur. Coordination can be also assimilated to search process performed in the distributed problems solving ([20], [7]). In this specific context, coordination refers to internal reasoning of agents, mechanisms of partial solutions exchanges, etc.

More recently, this background of cooperative agents has been discussed and coordination has been considered from a competitive way. The idea is that each agent has individual motivation, acts to achieve its own goal and interacts with other agents to ensure some global properties of the system. In [19], for example, Shoham *et al.* obtain coordination between agents having antagonist aims by introducing social laws. Their application deals with robots which have to move in an unknown environment. Since their

motivations are individual and their behavior not necessarily cooperative, conflicts can appear. To avoid such a situation, the authors introduce rules which allow to restrict possible actions for agents in conflict.

Crossing an intersection is a driving task which differs from one country to another. In the south of Europe, this task is mostly competitive especially for latin drivers. In the northern countries, the crossing is less competitive and more cooperative. The coordination mechanism has to be flexible in order to reproduce the large variety of drivers' behavior.

3.2 Multi-agent coordination in the ARCHISIM tool

The coordination mechanism used in ARCHISIM is based on the modeling of conflicts inside the intersection. This modeling applies a breaking down of the complex interactions between agents in elementary situations involving two agents.

An actual driver perceives a complex crossroad as a succession of elementary intersections of type T or X. A roundabout for example is a succession of T intersections. The interactions between drivers in an elementary intersection are regulated by priority relations defined in the Highway Code. Such a priority relation can be expressed by the predicate *prio*. In a X intersection, four interactions are possible and can be illustrated by the following conjunction:

1. $\neg prio(x, y) \wedge \neg prio(y, x)$: no conflict exists between the agents x and y
2. $prio(x, y) \wedge \neg prio(y, x)$: y comes from a minor road or has a sign and consequently has no priority over x
3. $\neg prio(x, y) \wedge prio(y, x)$: this situation is the dual of the previous situation
4. $prio(x, y) \wedge prio(y, x)$: x and y both have the priority, the figure 1 is an illustration of such a situation

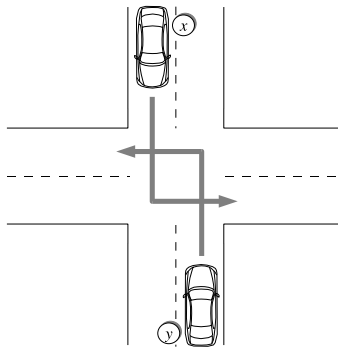


Figure 1: An elementary interaction between two agents at a crossroad

The coordination mechanism has the following dynamics. An agent near an intersection searches all vehicles with which it is in conflict. For each of them, the agent assesses the priority relation it has. Each priority relation is used as a local rule which indicates to the agent if it has to accelerate or slow down. When multiple priority relations are involved in its current situation, the agent chooses the behavior which induces the lower speed. The figure 2 is a synthesis of the coordination mechanism. For a more complete description of this coordination mechanism, the reader can refer to [3].

3.3 Priority evaluation

In order to obtain a valid overall traffic flow, the behavior of agents inside the intersection has to be as realistic as possible. The individual practices differ from one country to another. For example the Highway Code is barely respected by latin drivers and some psychology studies show that in other countries drivers tend to develop their own informal rules [2].

To take these features of the driving task into account, the agents have the possibility to alter the rules of the Highway Code with more informal rules during the second step of the coordination algorithm. For example, the following priorities can be considered by an agent:

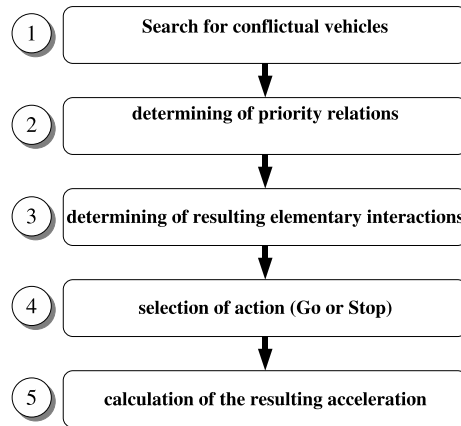


Figure 2: Steps of the coordination algorithm

- priority relating to speed: a driver arriving at an intersection and having priority (from the Highway Code point of view) tends to lose his priority if he perceives a vehicle approaching with a high speed
- priority relating to impatience: a driver stopped for a long time tends to consider he has priority on the other vehicles

The introduction of such abilities improves the local behavior of agents and consequently, by emergence, improves the global traffic obtained. In specific conditions such practices can lead to deadlocks inside the intersection. Indeed, the non-respect of several rules promotes the occupancy of the inner center of the crossroad but increases the risk of having a deadlock (figure 3).

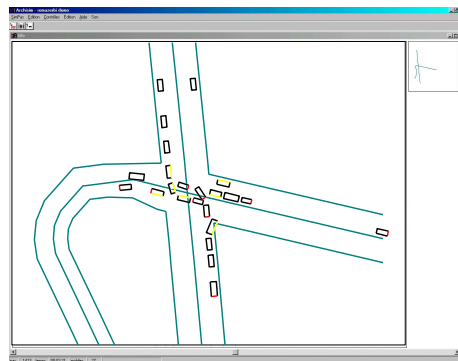


Figure 3: Example of deadlocks during a simulation

We claim that if agents have opportunistic behavior they also need anticipation abilities in order to avoid undesired situations like deadlocks.

4 Anticipation mechanism

4.1 Anticipation in artificial intelligence

Anticipation is a general concept and many definitions exist. Rosen's one is usually the most used in literature: *An anticipatory system is a system containing a predictive model of itself and / or of its environment that allows it to change current state at an instant in accord with the model predictions pertaining to a later instant* [18].

In the field of multi-agent system, anticipation is used to construct adaptive and complex behavior like in video game for instance [13]. In [5], the author proposes a multi-agent architecture for a special kind of anticipation: preventive anticipation which consists of using prediction of the future in order to prevent some undesired situations. The set of possible states of the system is divided in two parts: the desired and

```

procedure anticipate(in: ActionsList  $LA$ ,
                    UndesiredStatesList  $LU$ ,  $M(A, R, D)$ )
out: ActionsList  $LA$ )
begin
  propagation( $M$ )
  for each  $action \in LA$  do
     $r \leftarrow computeEffects(a)$ 
    updateRelation( $R, r$ )
    propagation( $M$ )
     $undesiredState \leftarrow undesiredStateSearch(M, LU)$ 
    if  $undesiredState$  then
       $LA \leftarrow LA - \{action\}$ 
    end
  end
end

```

Figure 4: Procedure *anticipate*

undesired states. In this context, anticipation is viewed as a process trying to keep the system in a desired state.

4.2 A framework for preventive anticipation based on constraints processing

We introduce a formal framework for preventive anticipation by using constraint processing techniques [12]. Our approach is based on the creation by each agent of a mental representation of environment and more especially of the different relations and interactions which exist between the agents of the environment [6].

A mental representation M of an agent x is a triplet (A, R, D) in which:

- $A = \{a_1, a_2, \dots, a_n\}$ is the subset of all agents perceived by x ,
- $R = \{r_1, r_2, \dots, r_k\}$ where each r_i is a binary relation expressing an interaction between two agents of A ,
- $D = \{dom(a_1), dom(a_2), \dots, dom(a_n)\}$ defining the domain of each agent in A .

The nature of the domain can diversify from an application to another. A domain can be for example spatial or temporal.

Based on this representation, each agent tries to predict the future state of the system, in other words it tries to determine how the actions of the agent will make the system evolve. In our framework of anticipation, we consider that each action has direct and indirect effects. A direct effect of an action is modeled by adding or deleting a relation in the set R . Indirect effects are computed by using propagation techniques over direct effects.

The aim of the anticipation algorithm is to scan a list of actions LA and to eliminate the actions which induce an undesired state. Undesired states are expressed as a domain list which is a parameter of the algorithm. The first step of the procedure *anticipate* starts by making a first propagation over M . This is performed by an external propagation function. Classic algorithm like AC-3 [16] can be used and adapted in accordance with the application. Then for each action of the list LA , direct effects are evaluated. This is also performed by an external function which can not be expressed in a generic manner since it depends on the application. In our traffic simulation example, this external function determines the direct effects with cinematic calculus on current position and speed of vehicles. Direct effects of an action are used to update the representation M . Based on this update, undesired states are searched. If an undesired state is found, the current action is removed from the list LA .

The function *undesiredStateSearch* scans the list LU and verifies that the undesired state it holds are not present in the mental representation M .

```

function undesiredStateSearch(in:  $M(A, R, D)$ ,
                               UndesiredStatesList  $LU$ )
begin
  for each  $\text{dom}(x) \in LU$  do
     $\text{dom}'(x) \leftarrow \text{domainSearch}(x, M)$ 
    if  $\text{dom}'(x) = \text{dom}(x)$  then
      return(true)
    end
  end
return(false)
end

```

Figure 5: Procedure *undesiredStateSearch*

5 Implementation and validation

5.1 Use of anticipation to avoid deadlocks

Here we present the use of our formal framework to avoid deadlocks when simulating traffic at junction in the ARCHISIM tool. To explain our model, we consider an agent x located at the entrance of a crossroad (figure 6).

The set A can be built with the agents present inside the intersection and perceived by x . The domain associated to each agent of A is temporal and expresses the next time step. For example, if an agent z has in its mental representation $\text{dom}(x) = [1, 4] \cup [8, 10]$, this means that z has inferred that x will be blocked during the interval $t + 5$ to $t + 7$.

The set R is composed with blocking relation. Three types are considered:

- $\text{bph}_z(a_x, a_y) \equiv$ “ a_x is physically blocked by a_y from the point of view of agent a_z ”
- $\text{bpha}_z(a_x, a_y) \equiv$ “ a_z anticipates that a_x will be physically blocked by a_y ”
- $\text{bpr}_z(a_x, a_y) \equiv$ “ a_y has priority over a_x from the point of view of agent a_z ”¹

Let us illustrate how the algorithm works on the example of the figure 6. Three vehicles a_t, a_s, a_z are stopped and blocked by a_y . The direction of each vehicle are the following:

- a_s comes from the north and wants to turn left,
- a_t comes from the east and wants to turn left,
- a_z comes from the east and wants to turn left,
- a_y comes from the west and wants to turn left,
- a_x comes from the south and wants to go straight on.

The agent a_x computes at the instant t the following representation:

$$\begin{aligned}
 A &= \{a_x, a_t, a_s, a_z, a_y\} \\
 D &= \{\text{dom}(a_x) = [1, +\infty[, \text{dom}(a_t) = [1, +\infty[, \\
 &\quad \text{dom}(a_s) = [1, +\infty[, \text{dom}(a_z) = [1, +\infty[, \\
 &\quad \text{dom}(a_y) = [1, +\infty[\} \\
 R &= \{\text{bpha}(a_x, a_t), \text{bph}(a_t, a_z), \text{bpr}(a_t, a_s), \\
 &\quad \text{bph}(a_s, a_z), \text{bph}(a_z, a_y), \text{bpha}(a_y, a_t), \\
 &\quad \text{prio}(a_x, a_y) \wedge \neg \text{prio}(a_y, a_x)\}
 \end{aligned}$$

The application of the anticipation algorithm gives the following reasoning:

¹The predicates *prio* and *bpr* both refer to a priority relation between agents. When an agent anticipates, it uses *prio* to express its own priority relations and *bpr* to express a priority relation between two other agents.

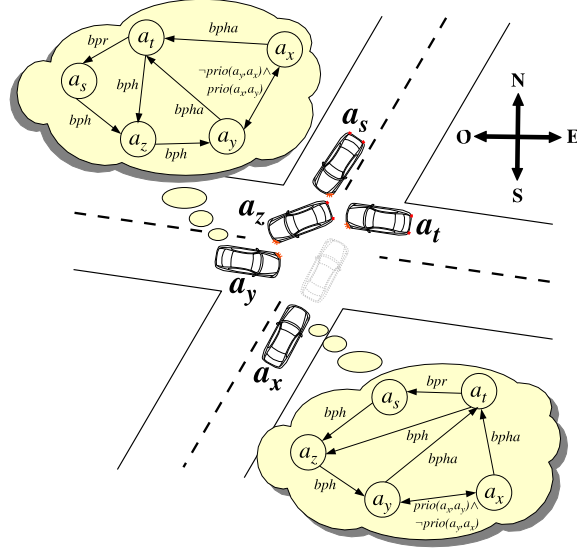


Figure 6: Mental representations of two agents in a crossroad

1. propagation: The domain of the five agents can be reduced. For example, considering that a_z is blocked by a_y , it is possible to deduce that a_z will remain brought to a standstill till a_y has not got past the conflict point. This time can be obtained with a cinematic calculation. If for instance this time is equal to 2 time steps, the values 1 and 2 can be deleted from the domain of a_z : $dom(a_z) = [3, +\infty[$. The same reasoning leads to simplify the domain of a_s and a_t : $dom(a_s) = [5, +\infty[$ and $dom(a_t) = [7, +\infty[$. The simplification of $dom(a_t)$ induces that $dom(a_y)$ can still be reduced since a relation between a_t and a_y exists: $bpha(a_y, a_t)$. Finally, the set D has been simplified and is now equal to:

$$D = \{dom(a_x) = [1, 5] \cup [9, +\infty[, dom(a_t) = [7, +\infty[, \\ dom(a_s) = [5, +\infty[, dom(a_z) = [3, +\infty[, \\ dom(a_y) = [1, 5] \cup [9, +\infty[\}$$

2. effects computation of actions of a_x : As the coordination algorithm only deals with longitudinal acceleration, two actions are possible for a_x : *Go* and *Stop*. If a_x chooses the action *Go*, its position on the road changes and consequently its distance with a_t is reduced. The blocking relation between a_x and a_t becomes effective: this is expressed in the mental representation by the replacement of the relation $bpha(a_z, a_t)$ by $bph(a_z, a_t)$. In the same way, the priority relation $prio(a_x, a_y) \wedge \neg prio(a_y, a_x)$ becomes a blocking relation: $bph(a_y, a_x)$. The set R is therefore updated as below:

$$R = \{bph(a_x, a_t), bph(a_t, a_z), bpr(a_t, a_s), \\ bph(a_s, a_z), bph(a_z, a_y), bpha(a_y, a_t), \\ bph(a_y, a_x)\}$$

3. propagation: Adding the relation $bph(a_y, a_x)$ allows to simplify once more the domain of a_y . But simplifying $dom(a_y)$ also implies the simplification of all domains of D . All domains can be reduced to an empty interval, which is coherent since all vehicles can no longer move:

$$D = \{dom(a_x) = \emptyset, dom(a_t) = \emptyset, dom(a_s) = \emptyset, \\ dom(a_z) = \emptyset, dom(a_y) = \emptyset\}$$

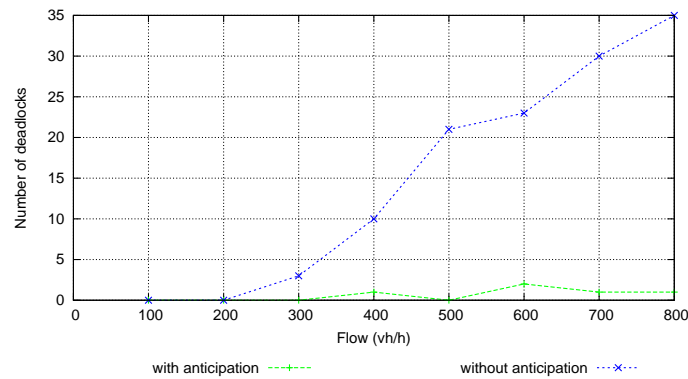


Figure 7: Variation of the number of deadlocks in accordance with the simulated flow

4. undesired states search: From a_x point of view, having its own domain empty can be considered as an undesired state. The agent a_x can eliminate the action $G\theta$ from its list of possible actions.

Finally, a_x has no other option and has to stop. The anticipation algorithm applied by a_y does not change its actions list. With the notion of impatience, it considers it has priority after a sufficient time and decides to accelerate.

6 Validation

The evaluation of our multi-agent model has been led in two parts. The first one results in tests and visual analysis of individual behaviors.

6.1 Experiments on individual behaviors

As the analysis of the individual behaviors is completely subjective, we use different scenarios allowing to place the agents in situations of traffic suitable for the appearance of deadlocks.

- signalized intersection with different traffic densities
- unsignalized intersection with different traffic densities
- sequences of double left-turn (with and without long vehicle)

The figure 7 shows how our anticipation algorithm reduces the number of deadlocks during the simulation of a non-signalized intersection. For a one hour simulation with non-anticipatory agents, the number of deadlocks varies from 0 to 35 according to the flow. When the agents have anticipation abilities, the number of deadlocks remains almost constant and close to zero for simulated flows going up to 800 vehicles/hour per axis.

To be more accurate in our evaluation, we have tried to simulate the traffic of a real crossroad and compare the traffic data obtained by simulation with real measures.

6.2 Simulation of a real junction

The intersection used for this evaluation is located in the city of Reggio (Calabria) in Italy. This intersection makes the junction between a main road going from north to south and a secondary street (from east to west). It is signalized by two stop signs installed on the secondary trunk road. Each arm of the junction has two lanes and the inner center is large enough to allow storage of vehicles.

The traffic data was measured manually by the University of Reggio Calabria within the intersection between 12:30 AM and 1:30 PM on a normal weekday. For each arm of the intersection, the entry flow is expressed in a number of vehicles per hour and percentage of turning movement (right or left turn, straight ahead).

From this data, we have generated a traffic demand which has been used in ARCHISIM to generate simulated vehicles two hundred meters before the intersection. Virtual sensor has been used to measure the flow of vehicles.

Simulations were carried out according to two experimental conditions. In the first experiment, only the coordination mechanism [3] (reference behaviour) is used, whereas the anticipation algorithm (anticipatory behaviour) is added in the second one.

The figures 8 to 11 draw the comparison between the simulated flows obtained with or without anticipation and the real flow measured. For all graphs, the time is expressed as five minutes' period. The figures 8 and 9 relate to the principal axis. The figures 10 and 11 describe the flows of the secondary axis.

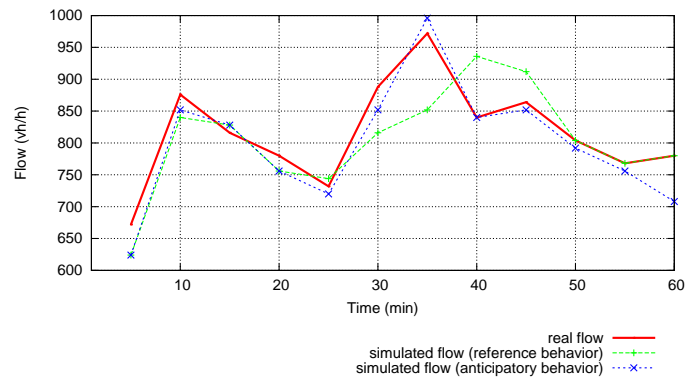


Figure 8: Simulated flow (North-South axis)

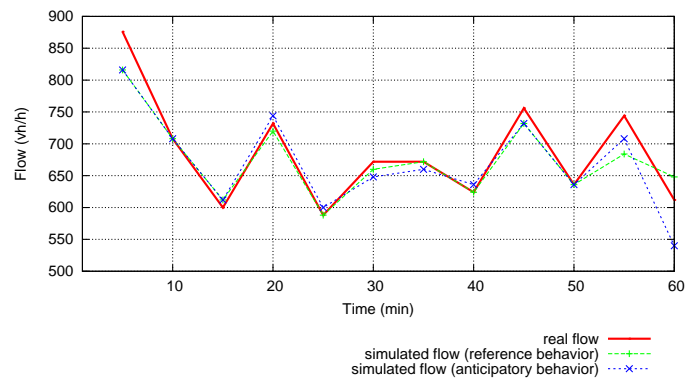


Figure 9: Simulated flow (South-North axis)

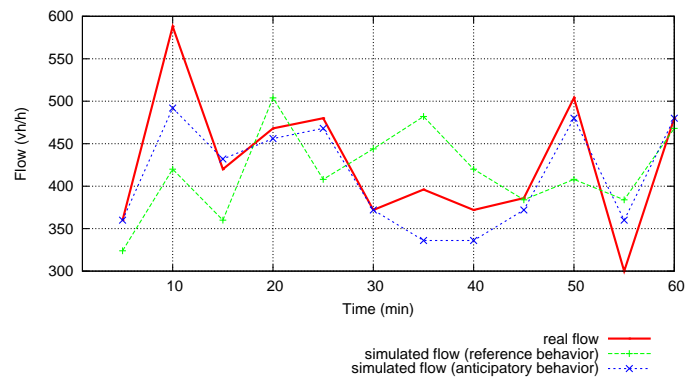


Figure 10: Simulated flow (East-West axis)

In all cases the best results are obtained by using anticipatory agents. To qualify more precisely these

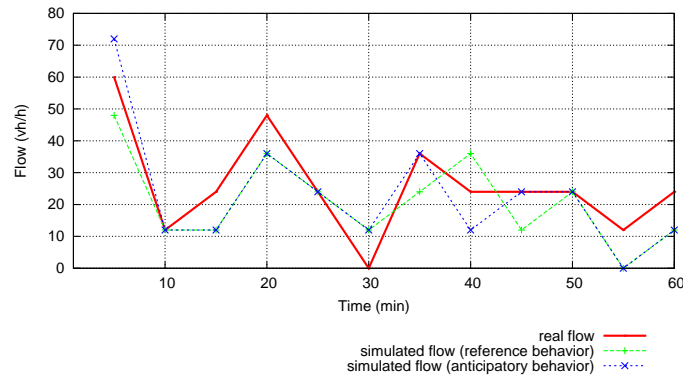


Figure 11: Simulated flow (West-East axis)

results, we use a well known indicator in traffic simulation: RMSE. It corresponds to the root mean square error between the real flow y_i and the flow simulated x_i and is expressed by the following formula:

$$RMSE = \sqrt{\frac{\sum_i (x_i - y_i)^2}{\sum_i y_i^2}}$$

The values obtained are, for three out of the four axis, lower than 0,1 what corresponds to an error lower than 10% (table 1). In the literature, a good simulation of traffic often offers an error rate lower than 15%. The results of the curve of the figure 11 are not very representative since the simulated flows are weak and not very significant.

	RMSE		Flow (vh/h)	
	reference behaviour	anticipatory behaviour	Min	Max
North/South	0,06	0,04	600	900
South/North	0,03	0,03	680	980
East/West	0,15	0,06	300	600
West/East	0,32	0,30	0	60

Table 1: Variation in RMSE with reference and anticipatory behaviour

7 Conclusion

In this article, we have presented the use of a multi-agent approach in the design of traffic simulation tool. In particular, we have introduced an original framework for anticipation in a multi-agent system based on constraint processing techniques.

This framework has been instantiated and used to avoid deadlocks when simulating traffic at junction. The resulting simulation tools have been evaluated by comparison with a real intersection. These experimental results confirm the relevance of the use of multi-agent techniques to develop a traffic simulation tool.

Researches are actually made to improve the realism of simulation at junctions. In particular, the multi-agent model will be extended to manage lateral acceleration and give more realistic behavior to simulated drivers.

References

- [1] Aimsun. *AIMSUN 5.0 Microsimulator User's Manual*. TSS-Transport Simulation Systems, 2005.
- [2] Gunilla M. Björklund and Lars Åberg. Driver behaviour in intersections: Formal and informal traffic rules. *Transportation Research Part F*, 8:239–253, 2005.

- [3] Alexis Champion, Stéphane Espié, René Mandiau, and Christophe Kolski. A game-based, multi-agent coordination mechanism - application to road traffic and driving simulations. In *Summer Computer Simulation Conference*, pages 644–649, Montréal, Québec, Canada, July 2003.
- [4] Alexis Champion, Ming-Yu Zhang, Jean-Michel Auberlet, and Stéphane Espié. Behavioural simulation: towards high-density network traffic studies. In *Proceedings of the Third International Conference on Traffic and Transportation Studies (ICTTS 2002)*, pages 988–995, July 2002.
- [5] Paul Davidsson. *Anticipatory Behavior in Adaptive Learning Systems*, chapter : A Framework for Preventive State Anticipation. Springer, 2003.
- [6] Arnaud Doniec, René Mandiau, Stéphane Espié, and Sylvain Piechowiak. Dealing with multi-agent coordination by anticipation: Application to the traffic simulation at junctions. In *Proceedings of the Third European Workshop on Multi-Agent Systems (EUMAS'05)*, Brussels, Belgium, December 2005.
- [7] Arnaud Doniec, Sylvain Piechowiak, and René Mandiau. A DisCSP solving algorithm based on sessions. In *Recent Advances in Artificial Intelligence: Proceedings of the 18th International Florida Artificial Intelligence Research Society Conference (FLAIRS'05)*, pages 666–670, Clearwater, USA, May 2005.
- [8] Sameh El Hadouaj, Alexis Drogoul, and Stéphane Espié. How to combine reactivity and anticipation: the case of conflicts resolution in a simulated road traffic. In *Multi-Agent-Based Simulation, Second International Workshop*, pages 82–96, Boston, USA, July 2000. Springer.
- [9] Stéphane Espié. Archisim, multi-actor parallel architecture for traffic simulation. In *Proceedings of the Second World Congress on Intelligent Transport Systems*, volume IV, Yokohama, Japan, November 1995.
- [10] Mickael P. Georgeff. Communication and interaction in multi-agent planning. In *Proceedings of the 3th National Conference on Artificial Intelligence (AAAI'83)*, pages 125–129, Washington, USA, August 1983.
- [11] Nicholas R. Jennings. Coordination techniques for distributed artificial intelligence. In *Foundations of Distributed Artificial Intelligence*, pages 187–210. Wiley, 1996.
- [12] Vipin Kumar. Algorithms for constraint-satisfaction problems: A survey. *Ai Magazine*, 13(1):32–44, 1992.
- [13] John E. Laird. It knows what you're going to do: Adding anticipation to a quakebot. In *Proceedings of Autonomous Agent'01*, pages 385–392, Montreal, Canada, 2001.
- [14] Victor R. Lesser and Daniel D. Corkill. Functionally accurate, cooperative distributed systems. In *IEEE Transactions on Systems, Man and Cybernetics*, pages 81–96, 1981.
- [15] Edward Lieberman and Ajay K. Rathi. *Traffic flow theory, expansion of the Transportation Research Board (TRB) Special Report 165*, chapter 10 - Traffic simulation, pages 10–1 10–25. Oak Ridge National Laboratory, 1997.
- [16] Alan K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8:99–118, 1977.
- [17] Thomas W. Malone. Modeling coordination in organizations and markets. *Management Sciences*, 33(10):1317–1332, 1987.
- [18] Robert Rosen. *Anticipatory Systems - Philosophical, Mathematical and Methodological Foundations*. Pergamon Press, 1985.
- [19] Yoav Shoham and Moshe Tennenholtz. On social laws for artificial agent societies: Off-line design. *Journal of Artificial Intelligence*, 1-2(73):231–252, 1995.
- [20] Makoto Yokoo. *Distributed Constraint Satisfaction: Foundations of Cooperation in Multi-agent Systems*. Springer Verlag, 2001.