DEALING WITH MULTI-AGENT COORDINATION BY ANTICIPATION: APPLICATION TO THE TRAFFIC SIMULATION AT JUNTIONS

Arnaud Doniec (1,2) and René Mandiau [1] and Stéphane Espié [2] and Sylvain Piechowiak [1]
[1] LAMIH, UMR 8530, 59313 Valenciennes
[2] INRETS, 94114 Arcueil

ABSTRACT

In contexts of competitive multi-agent coordination in a highly dynamic environment, one of the crucial problem is the resolution of deadlocks situations. In the context of multi-agent simulation, where the perception is often local, the detection of such situation is important. This article is devoted to the proposition of a model to select actions by anticipation. We detail our model for a simulation traffic application and in particular in critical situations such as crossroad.

INTRODUCTION

Behavioral simulation

Road traffic simulation is today used in many various applications: impact analysis of new road infrastructure [Espié *et al.*, 2002], driver support systems design and assessment [Auberlet *et al.*, 2002], etc. Two approaches of traffic simulation are possible. The mathematical approach reproduces vehicles stream from mathematical laws established with data collected in real situation. By opposite way, the behavioral approach produces an emerging traffic by reproducing interactions between various actors of the road situation (car driver, pedestrian, road operator, ...). The traffic observed in such a simulation model is therefore the result of the sum of all actors' individual actions. The behavioral approach is applied in a multi-agent framework: [Balmer *et al.*, 2004], [Sukthankar *et al.*, 1998], [Espié, 1995].

MSIS team from INRETS develops a behavioral simulation traffic model: ARCHISIM [Espié, 1995]. The computing model of ARCHISIM follows the multi-agent principles [Wooldridge, 2002]. Each simulated driver is an autonomous software agent which evolves in a virtual environment, and interacts with other agents of the simulation performing its goals according to its skills and the current situation. At each simulation step, a simulated driver receives information describing the surrounding situation in the environment. This description is quite complete and for example presents the other vehicles not only in the proximity of the driver but also in far areas. The computing model of ARCHISIM is supplied with a behavioral model of the driver based on results obtained in driving psychology [Saad, 1992]. Thus, the agents' behavior is not normative.

Simulation of urban traffic

The simulation of urban traffic is still badly processed in present simulation tools. In particular, intersections are considered in a simplistic way. An intersection is generally represented as a black box where a central process directly moves vehicles from the input to the output of the crossroad. In such a representation, there is not any conflicts management between vehicles inside the crossroad.

Traffic simulation at junction is a hard problem due to:

- the presence of conflictual streams: merging and intersection streams
- the potential complex geometry of the crossroad
- the inner-space of the intersection which is usually non-structured
- the side effect generated by an other near junction (up queue)
- the eventual deadlocks

As the driving task is competitive, traffic simulation at junction can be expressed as a non-cooperative multi-agent coordination problem. In this context, a game-based multi-agent coordination mechanism has been proposed in [Champion *et al.*, 2003] to coordinate simulated drivers at a simple and isolated crossroad.

In section 2, we briefly present this mechanism and analyse its relevance in the case of complex intersections. In particular, we show that it has to be completed by an anticipation mechanism to be correctly used when deadlocks appear. In section 3, a based on constraints model to anticipate is described. In section 4, we discuss the implementation of our model in ARCHISIM and give some preliminary results.

MULTI-AGENT COORDINATION AT SIMPLE INTERSECTION

As ARCHISIM considers the traffic from a distributed point of view, the mechanism needs to be distributed. Consequently, and as in actual situation, each agent has to interact with imperfect and incomplete information. Its behavior has to be non-normative since real drivers do not always respect the highway code depending on the situation. The number of these situations is so huge that it is impossible to integrate them into an exhaustive mechanism, the coordination has to be the most generic. The coordination mech-

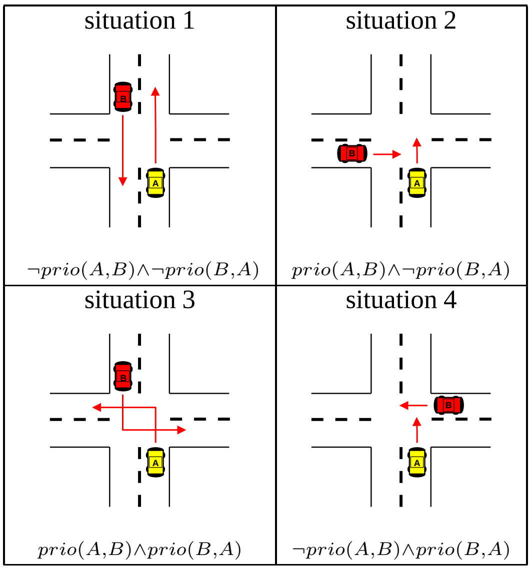

Figure 1: Elementary crossroad situations

situation 1	situation 2
$G_{A/B}=\begin{array}{cc} \begin{array}{cc} Go & Stop \end{array} & \\ \begin{pmatrix} (x_1,-z_1) & (x_3,z_3) \\ (0,0) & (0,0) \end{pmatrix} & \begin{array}{c} Go \\ Stop \end{array} \end{array}$	$G_{A/B}=\begin{array}{cc} \begin{array}{cc} Go & Stop \end{array} & \\ \begin{pmatrix} (-y_1,-y_2) & (y_3,y_6) \\ (0,0) & (0,0) \end{pmatrix} & \begin{array}{c} Go \\ Stop \end{array} \end{array}$
situation 3	**situation 4**
$G_{A/B}=\begin{array}{cc} \begin{array}{cc} Go & Stop \end{array} & \\ \begin{pmatrix} (x_1,-z_1) & (0,0) \\ (x_3,z_3) & (0,0) \end{pmatrix} & \begin{array}{c} Go \\ Stop \end{array} \end{array}$	$G_{A/B}=\begin{array}{cc} \begin{array}{cc} Go & Stop \end{array} & \\ \begin{pmatrix} (-y_2,-y_1) & (0,0) \\ (y_6,y_3) & (0,0) \end{pmatrix} & \begin{array}{c} Go \\ Stop \end{array} \end{array}$

$\{x_1,x_3,y_1,y_2,y_3,y_6,z_1,z_3\} \in \mathbb{N}^{+*}$

Figure 2: Matrix of elementary games

anism proposed in [Champion *et al.*, 2003] respects all these constraints contrary to other approaches such as [Trannois *et al.*, 1998].

Game-based coordination mechanism

The game theory provides a simple model to study conflictual situations between entities which share a same interest. It allows to study interactions between these entities. The game theory can be used to study coordination between agents [Beaufils, 2000] and to build multi-agent coordination mechanism [Rosenschein, 1986].

A game is a set of n players $\{j_1, j_2, ..., j_n\}$ in which for each j_i $(1 \leq i \leq n)$ is defined a set of actions $A_{j_i} = \{a_1, a_2, ..., a_m\}$ and a payoff fonction $G_{j_i} : A_{j_1} \times A_{j_2} \times \ldots \times A_{j_n} \rightarrow \mathbb{N}$ which associates a reward to each action. A solution of a game is a combination of actions allowing to have an optimal issue in term of winnings [Guerrien, 1997].

The game-based coordination mechanism used in ARCHISIM [Champion *et al.*, 2003] only manages the longitudinal acceleration, consequently the set of actions can be reduced to *Stop* and *Go*. At each time step, each agent analyses its vision and searches the vehicles which it is in conflict with. In this context, these vehicles are designed under the term of *players*. Two players share one of the four priority relations presented on figure 1.

The modeling proposed by Champion associates at each priority relation a 2-players game matrix. The general shape of this matrix is presented on figure 2. The details of the numerical application can be found in [Champion *et al.*, 2003].

An agent in conflict with n vehicles shares n priority relations and consequently can built n 2-players game matrix. To maximize its payoff, the agent has to aggregate its n 2-players matrix into a n-dimentional matrix of size 2^n. The maximisation is individual: a player sums all payoffs associated to the action Go and all payoffs associated to the action $Stop$ in the final aggregated matrix and chooses the biggest.

Analysis of the mechanism

As our problematic is to simulate complex traffic situations, we have analysed the relevance of the mechanism in this context. In particular, we have brought to the fore several characteristics of the mechanism showing that it cannot be directly applied to complex situations.

Firstly, we have underlined that the mechanism needs a relative stability concerning the choice of players and the perception of priorities. Since relations between simulated drivers evolve at each time step, the mechanism uses single shot games. This assumption can be acceptable, but coherence between games established between two time steps has to be ensured. For example, it seems that having a set of players completely different between two succesive time steps has no sense. Without stability, no convergence towards a global coordination of traffic is possible.

Secondly, considering only longitudinal acceleration do not allow to manage the storage areas which exist in crossroads. Simulated drivers inside a junction area, for example, are unable to change their lane. This constitutes a real limitation when considering complex junctions.

Finally, as the mechanism works with incomplete and imperfect information, deadlocks can appear. When the number of players grows, risks of having deadlocks grow too. In a 5-players game, only 40% of the global information are available for each agent and among all possible games issues, 30% can lead to deadlocks [Champion, 2003] (figure 3).

A part of these deadlocks can desappear between two time steps since relations between players are recomputed at each step of the simulation. Non-resolved deadlocks can lead to the complete lock of the intersection (figure 4).

Given this analysis, we assume that the coordination mechanism is not sufficient to deal with complex inter-

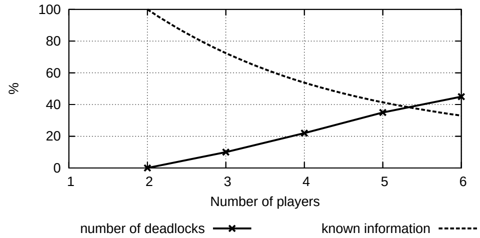

Figure 3: Efficiency of the mechanism [Champion, 2003]

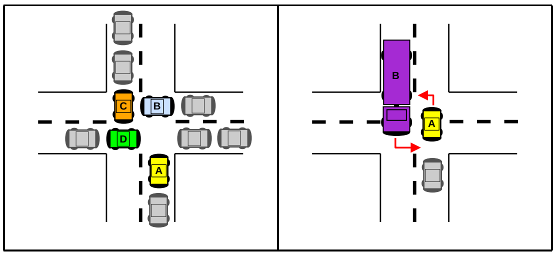

Figure 4: Two examples of deadlocks

sections. Two aspects have to be considered: anticipation and recognition of the context. From the psychological standpoint, and based on in-depth studies conducted in actual situation, Saad research works demonstrate the importance of anticipation in driving: "driving is anticipating" [Saad, 1992]. So the modeling of driver behavior must take into account anticipation particularly in complex road situations like intersections. From the multi-agent standpoint, the anticipation allows to built realistic and complex behavior [Laird, 2001].

ANTICIPATION LAYERS

The common definition used for anticipation in artificial system is: “An anticipatory system is a system containing a predictive model of itself and/or of its environment that allows it to change state at an instant in accord with the model predictions pertaining to a later instant” [Rosen, 1985].

Giving anticipation ability to an agent evolving in a dynamic environment requires :

- a representation of the environment
- a function to predict future states of the environment

The anticipation can be performed in two phases. A first phase consists in building the predictive model. The second phase is the interpretation of the results given by the predictive model.

Anticipating to prevent deadlocks

The deadlocks previously presented are the consequence of the non-anticipation of the agents. On the left example of the figure 4, the simulated driver A will play with the agent D. A perceives $prio(A, D) \wedge \neg prio(D, A)$ as priority relation and therefore builds the matrix $G_{A/D} = \begin{pmatrix} (-1,-4) & (2,0) \\ (0,1) & (0,0) \end{pmatrix}$. The sum of payoffs for the action Go is equal to 1 $(-1+2)$ which is superior to 0, the sum of payoffs for the action $Stop$. Consequently, the agent A chooses to go. This decision is effectively ’optimal’ at current time step but not at $t+1, 2, .., n$ since A will stick itself. As no one will be able to move, the deadlock will be infinite.

To avoid such infinite deadlocks, we propose to use a special kind of anticipation : preventive anticipation. This is based on anticipated undesired situations which make an agent change its strategy in order to avoid them. In our context, undesired situations are clearly infinite deadlocks.

In [Davidsson, 2003], the author presents preventive anticipation and introduces a general multi-agent architecture to practice it. He also proposes an instanciation of this architecture: linear anticipation.

Performing linearly anticipatory autonomous agents require making strong assumptions about the knowledge of the environment. Thus, it is necessary to act in a deterministic environment for which each agent has a quite complete representation. This requirement is in contradiction with our context where each simulated driver has to decide with incomplete and imperfect information.

One of the difficulties to use preventive anticipation in our context is to predict the future states of environment from the different strategies of agents. To simplify our problem we will consider, in a first time, that each agent is only able to anticipate about its own actions.

To predict consequences of its actions, an agent needs to evaluate its causal effects. To perform this evaluation, we distinguish two categories of effects: *local effect* and *global effect*. We call local effect the consequences that an action of an agent has over the other agents which share a direct relation with it. Global effect refers to consequences over agents which have no direct relation with A.

In most of problems, local effects of an action are simple to compute and can be quite reliable. The impact of the action over the global situation is more difficult to evaluate. Firstly, it is necessary to be able to evaluate correctly a part of the relations in which the agent is not directly involved. Secondly, a computation process is required to infer prediction about the global state according to 1) these evaluated relations and 2) the local effects mentioned above.

The evaluation of relations between two agents from the point of view of a tiers agent is a hard problem in general. In our application context, two types of relations can be perceived at an intersection: blocking relations and priority relations. The first ones are easy to evaluate since they only depend on the position and the speed of the vehicle. The seconds bring in the behaviors of the agents implicated in the relation, knowing that a tiers agent has no knowledge about their behaviors. Consequently, it has to make approximation about that they are going to do.

To infer prediction about global states, we introduce a formal approach of preventive anticipation based on constraints.

A formal model for preventive anticipation

The representation of the environment must contain information allowing to realize predictions of future states. For example, in our context: blocking and priority relations, distance between vehicles, speed and acceleration... As considered relations are all binary, a well fit representation can be constructed using a graph structure: each node symbolises a vehicle whereas edges express a relation between two vehicles. With a such representation, taking in account the local effect of an action can be realised by adding and/or deleting edges.

To evaluate the global effect easily, we choose to use a constraints network as graph stucture. A constraints network is a 3-uplet composed by a finite set of nodes $X = \{x_1, x_2, ..., x_n\}$ in which each x_i can take values in a domain $\mathsf{dom}(x_i)$ and is related to one or many binary constraints $c_k(x_i, x_j)$ $x_j \in X$.

A constraints network can have some inconsistances. An inconsistance can be defined as the impossibility to find a value for a node without violating any constraint. Our idea is to associate the non-desired states considered by the preventive anticipation framework as an inconsistance. To find potential inconsistances in the network, we use consistency technics and the associated propagation algorithm. To get more information about these technics, the reader can have a look to the following references: [Mackworth, 1977], [Kumar, 1992].

Our formalization is therefore defined as a constraints network where:

- nodes associate to noticed agents an abstract representation containing perceived information,
- constraints express binary relations between these agents,
- a domain represents the time interval during which the agent is able to act, in other words the complementary of the domain represents time intervals for which the agent is blocked.

From this formalization, we propose a general algorithm to anticipate. It uses a list of actions and a constraints network as arguments. For each action of the list, the algorithm determines if it will induce an undesired state in the future. To perform this, we use an auxiliary function to determine constraints corresponding to the local effect of actions (instruction `i4`). The algorithm returns the list of actions which do not create undesired states (instruction `i11`).

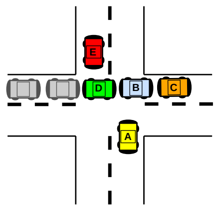

Figure 5: Example of relations occuring in a crossroad

```
function anticipate(ListOfActions LA,
                    ConstraintsNetwork CN)
begin
  propagate(CN);                            (i1)
  for each A in LA do
    copyOfCN <- CN;                         (i3)
    LC <- computeLocalEffectsOf(A);         (i4)
    AddConstraints(LC,CN);                  (i5)
    propagate(CN);                          (i7)
    if hasAnUndesiredState(CN)              (i8)
      then
        delete(A,LA)                        (i9)
    end;
    CN <- copyOfCN;                         (i10)
  end;
  return LA;                                (i11)
end
```

The complexity of our general algorithm depends on methods used for the propagation ([Bessière *et al.*, 1995], [Mackworth, 1977]) which complexity is bounded by $O(ed^2)$ and $O(ed^3)$ (d is the max length of the considered domains and e the total number of relations in the network). As this algorithm is run at each time step by each of the n agents involving in the situation, the global complexity is in the worst case $O(ned^2)$.

Application in ARCHISIM

Our model for preventive anticipation has been implemented in ARCHISIM in order to avoid infinite deadlocks. To construct a representation of a situation occuring at a crossroad, we use two types of relations: blocking and priority relations. About the blocking ones, we consider two subtypes: effective and anticipated ones. To illustrate our speech, let us consider the figure 5.

Two vehicles D and B are jamed into the intersection. In our representation, we will say that B is *physically blocked* by D. A vehicle C comes behind B. It is not yet jamed, but according to the context we can deduce that C will be jamed in 2 or 3 time steps. We will say that C is *physically blocked with anticipation* by B. The same reasoning can be done for the vehicle A: A is physically blocked with anticipation by B. As C is coming on its right, we will consider in our representation the following relation: C has priority over A.

Evaluating this priority relation is not simple from the point of view of E. Indeed, E do not know how C and A perceive this relation and consequently can not know the games they will play. To anticipate, E has to make assumptions over the behaviors of C and A. Two solutions are possible. E can try to project its own behavior in the situation of C or A. This approximation is very subjective. Another solution is to consider that A and C will act with a normative behavior (as far as the highway code is concerned): C will go before A.

So, we use 3 relations: bph, $bpha$ and bpr to which we give the following semantics:

- $bph_z(x, y) \equiv$ *"x is physically blocked by y from the point of view of agent z"*
- $bpha_z(x, y) \equiv$ *"z perceives that x will be physically blocked by y"*
- $bpr_z(x, y) \equiv$ *"y has priority over x from the point of view of agent z"*

Each node represents a vehicle perceived by an agent which is taking a decision. To each node is associated a domain, it represents the next simulation time steps of the agent and is expressed as a set of integers. At instant t, if a value x is not present in the domain of an agent, this means that it will be jamed at $t + x$. For example:

- $dom(x) = [1...+\infty] \equiv$ *"x can potentially move during the interval* $t + 1$ *to* $+\infty$*"*
- $dom(x) = [1...4] \cup [8...10] \equiv$ *"x is blocked from* $t + 5$ *to* $t + 7$*"*

In a chain of blocked vehicles, the end of the chain can not move before the beginning. To evaluate the arc-consistance of our three relations, we can map them as Allen's relations [Allen, 1991]:

$$\begin{aligned}
bph_z(x, y) &\equiv x \texttt{ After } y \\
bpa_z(x, y) &\equiv (x \texttt{ After } y) \vee (x <= ttc(x, y)) \\
bpr_z(x, y) &\equiv x \texttt{ After } y
\end{aligned}$$

ttc stands for the time to conflict between x and y, ie in our case the time for x to cover the distance $|xy|$.

Preliminary results

To illustrate our proposal, let us consider a congested traffic situation on a X-crossroad. Fourteen simulated vehicles are jamed at the junction. The vehicle number 13 is getting closer to the intersection and the vehicle 8 is stopped inside the crossroad. The first thumbnail of the figure 6 illustrates this scenario.

In such a situation, the agent 13 chooses 8 as player since all vehicles are stopped and consequently can not play. Agent 8 follows the same reasoning. The situation is therefore modeled by the two matrix: $prio(13, 8) \wedge \neg prio(8, 13)$ from the standpoint of agent 13 and $\neg prio(8, 13) \wedge prio(13, 8)$ from the standpoint of agent 8. The maximization of payoffs on these two matrix game respectively gives Go and $Stop$ to agents 13 and 8. This process is itered from the time step $t = 1$ to $t = 270$ (figure 6). At step $t = 270$, the vehicle 13 can no more go, it has stuck itself by creating an infinite deadlock.

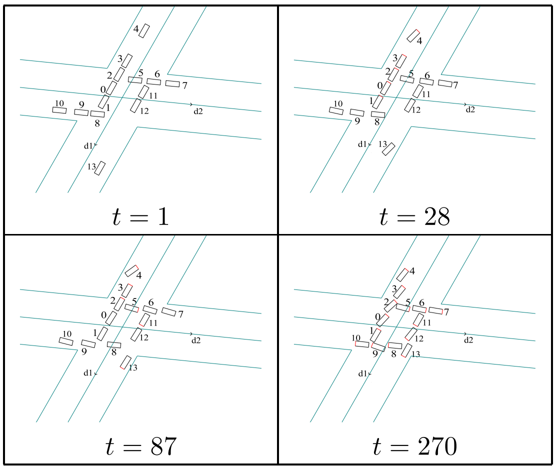

Figure 6: Sequences of the simulation without anticipation in ARCHISIM

The figure 7 presents agents having anticipation abilities in the same situation. To make our explanation, we will consider the simulated driver 13. At time step $t = 1$, the agent 13 starts its decision process by constructing a mental representation of the situation.

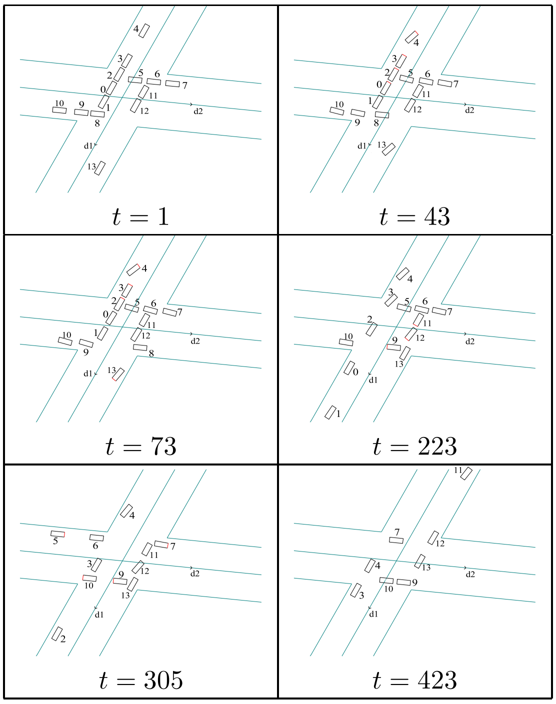

Figure 7: Sequences of the simulation with anticipation in ARCHISIM

To perform this, it can firstly consider all vehicles which are physically stuck. These vehicles make up a chain of jam and can be expressed by: $bpha(13, 12) \wedge bph(12, 11) \wedge bph(11, 6) \wedge ... \wedge bph(0, 1) \wedge bph(1, 8)$. Then, the agent 13 can add other perceived agents which do not belong to this chain. For example, by considering the vehicle 10, our agent can add other relations: $bpr(10, 0)$, $bpr(10, 2)$, etc.

To complete the construction of the constraints network, the agent number 13 associates a domain to each considered vehicle. At the beginning of the algorithm, all domains are initialized with $[1... + \infty]$. After the first propagation (instruction `i1`), the agent 13 has an approximation of the interval of blocking time of each vehicle including itself: $\mathsf{dom}(13) = [1..4] \cup [20... + \infty]$.

According to its position, its speed and its acceleration, the agent 13 will determine its future position. With this new position, it will compute a new vision in order to know the new topologic relations that this moving entails. Then, these topologic relations are interpreted as blocking or priority relations. All this process fit in with the instruction `i4` in the general algorithm.

The new obtained constraints are added to the network (some existing relations can also be updated or deleted if they are out-of-time) and a new propagation is performed (instruction `i7`). The agent 13 can therefore search undesired states (instruction `i8`). In particular, it can notice that its domain has become: $\mathsf{dom}(13) = \varnothing$ which constitutes an infinite deadlock for itself. Consequently, it deletes the action Go from the list of its available actions (instruction `i9`). In the end, the agent 13 chooses to stop.

The figures 8 and 9 respectively present the speed and acceleration resulting curves for the two previously presented situations. We can especially visualise the effect of anticipation on the speed of the agent 13 between the step 0 and 100. It is this strong deceleration which allows the agent 8 to detect that it has finally priority over 13.

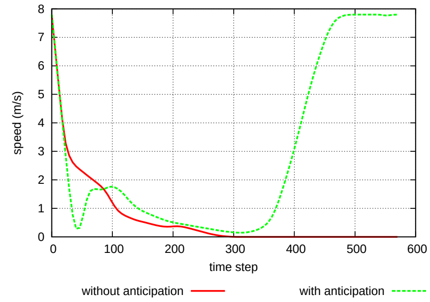

Figure 8: Speed curve of agent 13

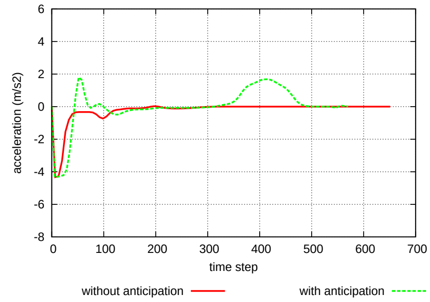

Figure 9: Acceleration curve of agent 13

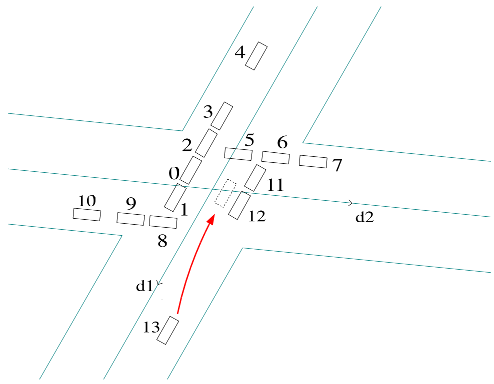

Figure 10: Example of space management in the center of the intersection

FUTUR WORKS

The next step of our work is to extend our model to manage space. Our idea is to join a composant for lateral acceleration to the two strategies Go and $Stop$. It could for instance Go_on_left and Go_on_right. In the example of figure 10, the vehicle 13 could choose to enter into the crossroad to stock itself in the inner-space of the intersection.

By improving this important aspect of the driving task, we hope to validate complex traffic situations on a succession of intersections from real data.

CONCLUSION

Through this article, we have presented the behavioral simulation model: ARCHISIM and its multi-agent approach of the traffic coordination at intersections. We have shown that the games based coordination mechanism is not enough to deal with complex traffic situations. Our proposition is to add anticipatory abilities to simulated drivers through a constraints-based on approach. We have given a general framework including an algorithm to perform preventive anticipation. This work has been implemented in ARCHISIM in order to avoid infinite deadlocks at junctions. As our approach is quite generic, we plan to use it for the vehicles space management in the inner-space of the crossroad. All these improvements should give a more realistic traffic from a visual and statistical point of view.

REFERENCES

Allen, 1991. ALLEN, J. (1991). Time and time again : The many ways to represent time. *Journal of Intelligent Systems*.

Auberlet *et al.*, 2002. AUBERLET, J., ESPIÉ, S. et CHAMPION, A. (2002). Study of the impact of an autonomous alert system by messages with the behavioral simulation model archisim. *In World Congress on Intelligent Transport Systems*, Chicago, USA.

Balmer *et al.*, 2004. BALMER, M., CETIN, N., K. et RANEY, B. (2004). Towards truly agent-based traffic and mobility simulations. *In Proceedings of AAMAS'04*, New-York, USA.

Beaufils, 2000. BEAUFILS, B. (2000). *Modèles et simulations informatiques des problèmes de coopération entre agents*. Thèse de doctorat, Université des Sciences et Technologies de Lille.

Bessière *et al.*, 1995. BESSIÈRE, C., FREUDER, E. et RÉGIN, J. (1995). Using inference to reduce arc consistency computation. *In Proceeding of IJCAI'95*, pages 592–598, Montréal, Canada.

Champion, 2003. CHAMPION, A. (2003). *Mécanisme de coordination multi-agent fondé sur les jeux: application à la simulation comportementale de trafic routier en situation de carrefour*. Thèse de doctorat, Université de Valenciennes.

Champion *et al.*, 2003. CHAMPION, A., ESPIÉ, S., MANDIAU, R. et KOLSKI, C. (2003). 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.

Davidsson, 2003. DAVIDSSON, P. (2003). *Anticipatory Behavior in Adaptive Learning Systems*, chapitre A Framework for Preventive State Anticipation. Springer.

Espié, 1995. ESPIÉ, S. (1995). Archisim, multi-actor parallel architecture for traffic simulation. *In Proceedings of the Second World Congress on Intelligent Transport Systems*, Yokohama, Japan.

Espié *et al.*, 2002. ESPIÉ, S., AUBERLET, J. et ZHANG, M. (2002). Approche intégrée pour l'étude de nouveaux profils routiers. *In Driving Simulation Conference 2002*, pages 175–194.

Guerrien, 1997. GUERRIEN, B. (1997). *La théorie des jeux*. Paris: Economica.

Kumar, 1992. KUMAR, V. (1992). Algorithms for constraint-satisfaction problems : A survey. *Ai Magazine*.

Laird, 2001. LAIRD, J. (2001). It kwows what you're going to do: Adding anticipation to a quakebot. *In Proceeding of Autonomous Agent 2001*, Montreal, Canada.

Mackworth, 1977. MACKWORTH, A. (1977). Consistency in networks of relations. *Artificial Intelligence*.

Rosen, 1985. ROSEN, R. (1985). *Anticipatory Systems - Philosophical, Mathematical and Methodological Foundations*. Pergamon Press.

Rosenschein, 1986. ROSENSCHEIN, J. (1986). *Rational Interaction: Cooperation Among Intelligent Agents*. Thèse de doctorat, Stanford University.

Saad, 1992. SAAD, F. (1992). In-depth analysis of interactions between drivers and the road environment : contribution of on-board observations and subsequent verbal report. *In Proceedings of the 4th Workshop of ICTCT*, University of Lund.

Sukthankar *et al.*, 1998. SUKTHANKAR, R., BALUJA, S. et HANCOCK, J. (1998). Multiple adaptive agents for tactical driving. *International Journal of AI*.

Trannois *et al.*, 1998. TRANNOIS, H., LEBRUN, A. et DELEAGE, J. (1998). A multi-agent framework for car traffic simulation. *In The Third International Conference and Exhibition on the Practical Application of Intelligent Agents and Multiagent Technology PAAM 98*.

Wooldridge, 2002. WOOLDRIDGE, M. (2002). *Introduction to MultiAgent Systems*. John Wiley & Sons.