

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

A. Doniec ^{ab} S. Espié ^b R. Mandiau ^a S. Piechowiak ^a

^a *LAMIH, Université de Valenciennes, 59400 Valenciennes Cedex 9*

^b *INRETS, 2 avenue du Général Malleret de Joinville, 94110 Arcueil*

Abstract

In contexts of competitive multi-agent coordination in a highly dynamic environment, one of the crucial problems is the resolution of deadlocks situations. In the field of multi-agent simulation, such situations can appear owing to the local perception of each agent. This article is dedicated to the proposition of a model actions selection by anticipation. We detail this model within the framework of a traffic simulation application and show its relevance for critical situations such as traffic jam in crossroad.

1 Introduction

1.1 Behavioral simulation

Road traffic simulation is used in many various applications nowadays: impact analysis of new road infrastructure [9], driver support systems design and assessment [10], etc. Two approaches of traffic simulation are possible. The mathematical approach reproduces vehicles streams from mathematical laws established with collected data of real situations. By opposite way, the behavioral approach produces an emerging traffic by managing interactions between various actors of the road situation (car driver, pedestrian, road operator, ...). The traffic, observed in such a simulation model, is therefore the sum of all actors' individual actions. The behavioral approach can be applied in a multi-agent framework: [2], [20], [8]. MSIS team from INRETS developed a behavioral simulation traffic model: ARCHISIM [8]. The computing model of ARCHISIM follows the multi-agent principles [21]. 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, we call "*vision*" 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 [19]. Thus, the agents' behaviors are not normative like in real life.

1.2 Simulation of urban traffic

The simulation of traffic in an urban network is still badly processed in present simulation tools. In particular, junctions 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 conflictual streams: merging and intersection (figure 1)

- the possible complex geometry of the crossroad
- the inner space of the intersection which is usually non-structured
- the side effects generated by an other near junction (queue of vehicles stretching away)
- the possible deadlocks

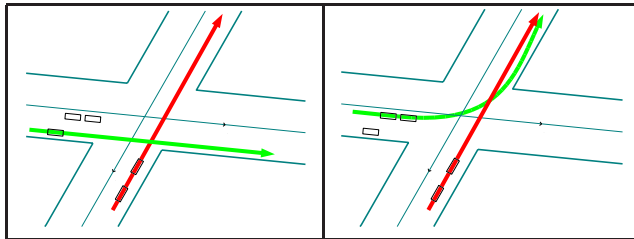


Figure 1: Conflictual streams at junctions

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 [6] and implemented in ARCHISIM in order to coordinate simulated drivers in 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 highlight the inability of algorithm to avoid complex deadlocks. In order to fill this lack, we propose to complete it by an anticipation mechanism. The formal model of this mechanism is introduced in section 3. We discuss its implementation in ARHCISIM and give some preliminary results.

2 Multi-agent coordination at simple intersection

The usual coordination mechanisms in litterature do not fulfill the constraints raised by our problematic.

As ARCHISIM considers the traffic from a distributed point of view, the coordination mechanism is also distributed. Consequently, and as in actual situation, each agent has to interact with imperfect and incomplete information.

The mechanism has to product non-normative behaviors since real drivers do not always respect the highway code depending on the situation. This condition imply the mechanism has to be generic since the number of driving situations is so huge that it is impossible to integrate them into an exhaustive mechanism.

The mechanism proposed in [6] has been specially designed for respecting all characteristics mentionned bellow. It's based on an adaptation of game theory.

2.1 Game-based coordination mechanism

The game theory provides a general framework to formalize conflictual situations between entities which share a same interest. In artificial intelligence, the game theory can be used to study coordination between agents [3] and to build multi-agent coordination mechanism [18].

A game is a set of n players $\{j_1, j_2, \dots, j_n\}$ in which for each j_i ($1 \leq i \leq n$) is defined a set of actions $A_{j_i} = \{a_1, a_2, \dots, a_m\}$ and a payoff function $G_{j_i} : A_{j_1} \times A_{j_2} \times \dots \times A_{j_n} \rightarrow \mathbb{N}$ 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 [12].

The game-based coordination mechanism used in ARCHISIM [6] 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 (ie the set of information describing its environment) 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 2.

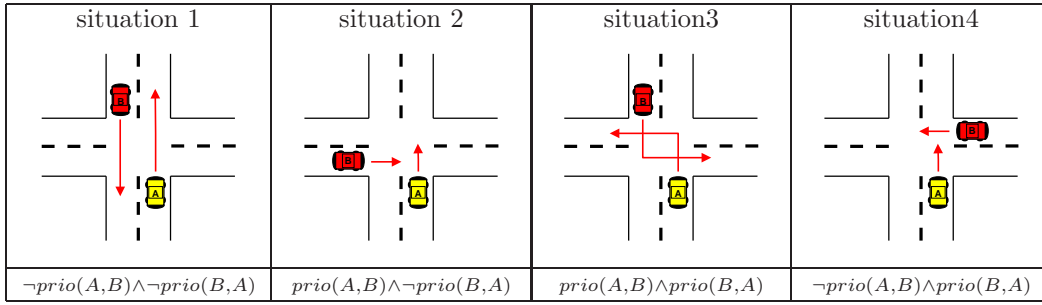


Figure 2: Elementary crossroad situations

The modeling proposed in [5] associates at each priority relation a 2-players game matrix. The general shape of this matrix is presented on figure 3. The details of the simplifications and numerical applications can be found in [6].

<p>situation 1</p> $G_A = \begin{pmatrix} Go & Stop \\ x_1 & x_3 \\ 0 & 0 \end{pmatrix} \begin{matrix} Go \\ Stop \end{matrix}$ $G_B = \begin{pmatrix} Go & Stop \\ -z_1 & z_3 \\ 0 & 0 \end{pmatrix} \begin{matrix} Go \\ Stop \end{matrix}$	<p>situation 2</p> $G_A = \begin{pmatrix} Go & Stop \\ -y_1 & y_3 \\ 0 & 0 \end{pmatrix} \begin{matrix} Go \\ Stop \end{matrix}$ $G_B = \begin{pmatrix} Go & Stop \\ -y_2 & y_6 \\ 0 & 0 \end{pmatrix} \begin{matrix} Go \\ Stop \end{matrix}$	<p>situation 3</p> $G_A = \begin{pmatrix} Go & Stop \\ x_1 & 0 \\ x_3 & 0 \end{pmatrix} \begin{matrix} Go \\ Stop \end{matrix}$ $G_B = \begin{pmatrix} Go & Stop \\ -z_1 & 0 \\ x_3 & 0 \end{pmatrix} \begin{matrix} Go \\ Stop \end{matrix}$	<p>situation 4</p> $G_A = \begin{pmatrix} Go & Stop \\ -y_2 & 0 \\ y_6 & 0 \end{pmatrix} \begin{matrix} Go \\ Stop \end{matrix}$ $G_B = \begin{pmatrix} Go & Stop \\ -y_1 & 0 \\ y_3 & 0 \end{pmatrix} \begin{matrix} Go \\ Stop \end{matrix}$
$\{x_1, x_3, y_1, y_2, y_3, y_6, z_1, z_3\} \in \mathbb{N}^{+*}$			

Figure 3: Matrix of elementary games

An agent in conflict with n vehicles shares n priority relations and consequently can build n 2-players game matrix. To maximize its payoff, the agent has to aggregate its n 2-players matrix into a n -dimensional 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 [5].

2.2 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, the mechanism needs a relative stability concerning the choice of players and the perception of priorities. Indeed, since relations between simulated drivers evolve at each time step, the mechanism uses single shot games. This implies that a continuity must exist between games computed during a time interval. For example, having a set of players completely different between two successive time steps is not very realistic. This aspect is very important because, 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. For example, simulated drivers inside a junction area 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 [5] (figure 4).

A part of these deadlocks can disappear between two time steps because relations between players are recomputed at each step of the simulation. Non-resolved deadlocks can lead to the complete lock of the intersection (figure 5).

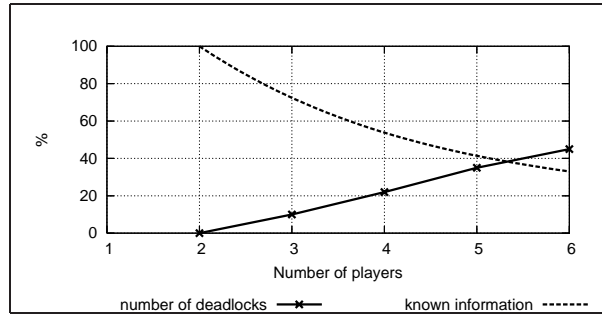


Figure 4: Efficiency of the mechanism [5]

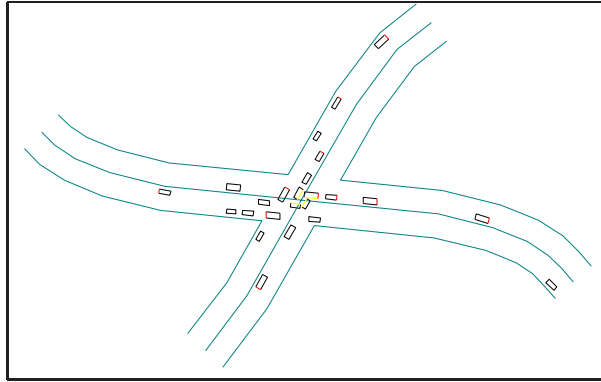


Figure 5: Examples of deadlocks in ARCHISIM

Given this analysis, we assume that the coordination mechanism is not sufficient to deal with complex intersections. 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's research works demonstrate the importance of anticipation in driving: "driving is anticipating" [19]. 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 constitute a promising research orientation which is, for example, used to built more realistic and complex behavior [14].

3 Anticipation layers

Anticipation is a general concept. First work about anticipation have been led by biologist and psychologist which try to explain the adaptative behavior of some animals.

In 1985, Robert Rosen propose for the first time a general framework of anticipation. The definition, he introduced, makes the link between knowledge of future and decision making at present time: "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" [17].

3.1 Anticipation in multi agent system

In the domain of multi-agent system, anticipation is used in various application like the creation of complex and adaptative behavior in video games [14] or the planification in high dynamic environment [11].

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.

One of the difficulties in anticipation is the utilisation of prediction to modifiante the behavior of an agent. In [16], the author introduces a simple ways to perform this. The main idea is to split the world of the agent into two parts: the desired and undesired states. Thus, the anticipation can be express as a process which try to avoid the undesired states.

In [7], the author presents an implementation of this principle and introduces a general multi-agent architecture to practice it. He also proposes an instantiation of this architecture: linear anticipation. In particular, he shows that 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.

3.2 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 5, 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 $t + 1, 2, \dots, n$ 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 consider undesired situations as infinite deadlocks. 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 usually a hard problem. 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 ones bring in the behavior 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 what they are going to do.

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

3.3 A formal model for anticipation

The representation of the environment must contain information allowing to perform 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 structure. A constraints network is a 3-uplet composed by a finite set of nodes $X = \{x_1, x_2, \dots, x_n\}$ in which each x_i can take values in a domain $\text{dom}(x_i)$ 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 inconsistencies. An inconsistency 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 inconsistency. To find potential inconsistencies 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: [15], [13].

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).

```

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 ([4], [15]) 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 $O(ned^2)$.

3.4 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 occurring 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 6.

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

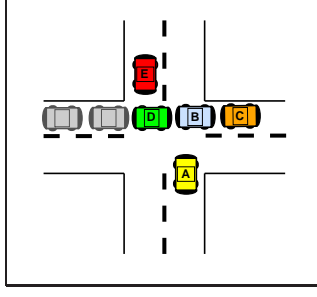


Figure 6: Example of relations occuring in a crossroad

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 [1]:

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

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

3.5 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 7 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 iterated from the time step $t = 1$ to $t = 270$ (figure 7). At step $t = 270$, the vehicle 13 can no more go, it has stuck itself by creating an infinite deadlock.

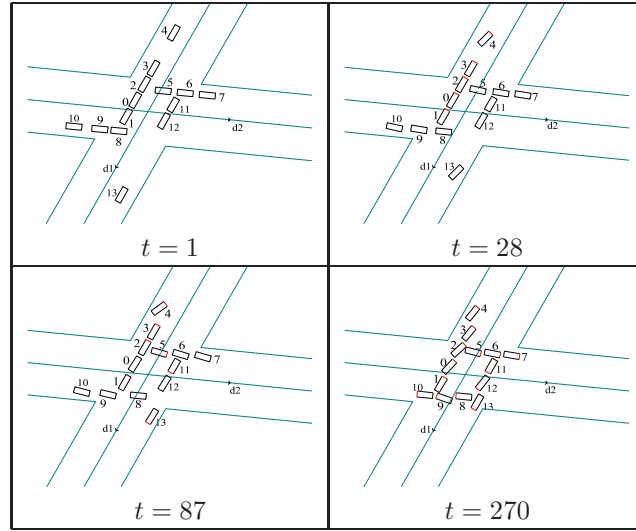


Figure 7: Sequences of the simulation without anticipation in ARCHISIM

The figure 8 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.

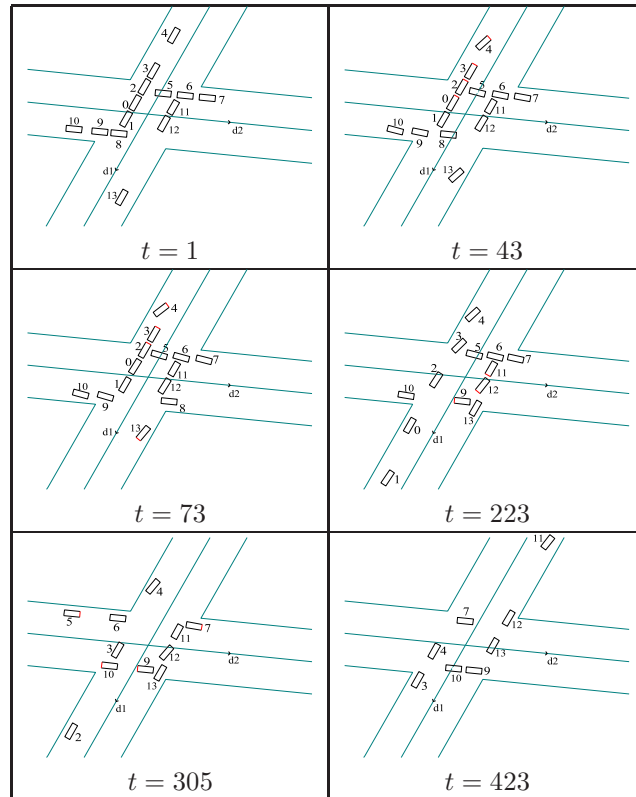


Figure 8: Sequences of the simulation with anticipation in ARCHISIM

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: $bpha(13, 12) \wedge bph(12, 11) \wedge bph(11, 6) \wedge \dots \wedge bph(0, 1) \wedge bph(1, 8)$. 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: $dom(13) = [1..4] \cup [20... + \infty]$.

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: $dom(13) = \emptyset$ 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 9 and 10 respectively present the speed and acceleration resulting curves for the two previously presented situations. We can especially visualize 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.

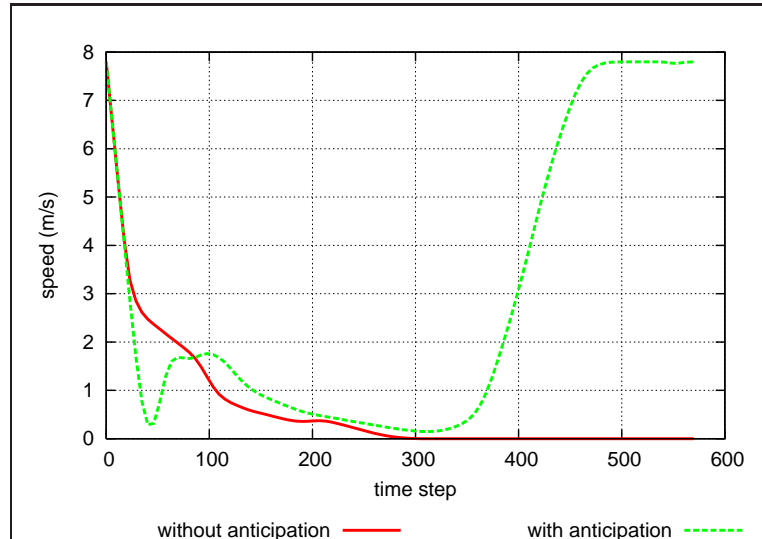


Figure 9: Speed curve of agent 13

4 Futur works

The next step of our work is to extend our model to manage space. Our idea is to join a component 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 11, 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.

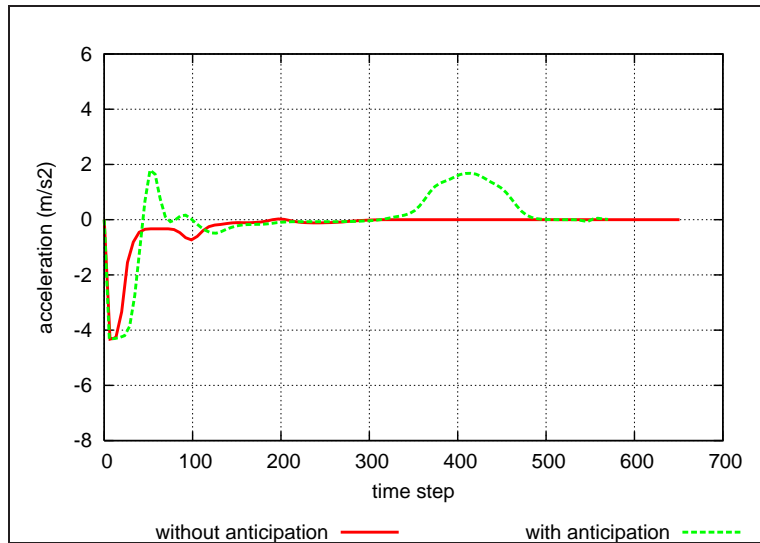


Figure 10: Acceleration curve of agent 13

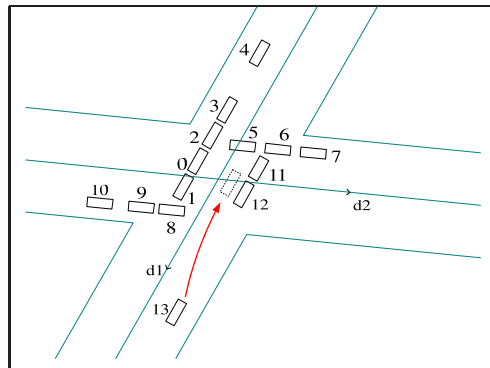


Figure 11: Example of space management in the center of the intersection

5 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 a 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 cross-road. All these improvements should give a more realistic traffic from a visual and statistical point of view.

References

- [1] J. Allen. Time and time again: The many ways to represent time. *Journal of Intelligent Systems*, 1991.
- [2] M. Balmer, N. Cetin, and B. Raney. Towards truly agent-based traffic and mobility simulations. In *AAMAS'04*, New-York, USA, 2004.
- [3] B. Beaufils. *Modèles et simulations informatiques des problèmes de coopération entre agents*. PhD thesis, Université des Sciences et Techniques de Lille, 2000.

- [4] C. Bessière, E. Freuder, and J.C. Regin. Using inference to reduce arc consistency computation. In *Proceedings of the International Conference on Artificial Intelligence*, pages 592–598, Montréal, 1995.
- [5] A. Champion. *Mécanisme de coordination multi-agent fondé sur des jeux : application la simulation comportementale de trafic routier en situation de carrefour*. PhD thesis, Université de Valenciennes et du Hainaut-Cambrésis, 2003.
- [6] A. Champion, S. Espié, R. Mandiau, and C. 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, juillet 2003.
- [7] P. Davidsson. *Anticipatory Behavior in Adaptive Learning Systems*, chapter : A Framework for Preventive State Anticipation. Springer, 2003.
- [8] S. Espié. Archisim, multi-actor parallel architecture for traffic simulation. In *Proceedings of the Second World Congress on Intelligent Transport Systems*, Yokohama, Japan, 1995.
- [9] S. Espié, J. Auberlet, and M. Zhang. Approche intégrée pour l'étude de nouveaux profils routiers. In *Driving Simulation Conference 2002*, pages 175–194, 2002.
- [10] S. Espié and A. Champion. 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, 2002.
- [11] I. Degirmenciyan-Cartault F. Marc, A. El Fallah-Seghrouchni. Modelling, control and validation of multi-agent plans in highly dynamic context. In *AAMAS'04*, New-York, USA, 2004.
- [12] B. Guerrien. *La théorie des jeux*. Paris: Economica, 1997.
- [13] V. Kumar. Algorithms for constraint-satisfaction problems : A survey. *Ai Magazine*, 1992.
- [14] J.E Laird. It knows what you're going to do: Adding anticipation to a quakebot. In *Proceeding of Autonomous Agent 2001*, Montreal, Canada, 2001.
- [15] A.K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 1977.
- [16] R. Rosen. Planning, management, policies and strategies: Four fuzzy concepts. *International Journal of General Systems*, 1974.
- [17] R. Rosen. *Anticipatory Systems - Philosophical, Mathematical and Methodological Foundations*. Pergamon Press, 1985.
- [18] J. Rosenschein. *Rational Interaction: Cooperation Among Intelligent Agents*. PhD thesis, Stanford University, 1986.
- [19] F. Saad. 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, 1992.
- [20] R. Sukthankar, S. Baluja, and J. Hancock. Multiple adaptive agents for tactical driving. *International Journal of AI*, 1998.
- [21] M. Wooldridge. *Introduction to MultiAgent Systems*. John Wiley & Sons, 2002.