ADAPTIVE BEHAVIOUR FOR PRISONER DILEMMA STRATEGIES BASED ON AUTOMATA WITH MULTIPLICITIES

Cyrille Bertelle, Marianne Flouret, Véronique Jay,
Damien Olivier and Jean-Luc Ponty
Laboratoire d'Informatique du Havre
25 rue Philippe Lebon, BP 540
76058 Le Havre Cedex - France
E-mail: {Cyrille.Bertelle, Marianne.Flouret, Veronique.Jay,
Damien.Olivier, Jean-luc.Ponty}@univ-lehavre.fr

KEYWORDS

agent, automaton, behaviour, organization, multiplicities, genetic algorithm.

ABSTRACT

This paper presents the use of automata with multiplicities for the modelization of agent behaviours. Genetic algorithms can be defined on their probabilistic forms and are able to simulate adaptive behaviours. An application to a simulation of evolutive strategies in game theory is presented, specifically for the prisoner dilemma which is considered as a basic model of dynamic agents interactions in terms of cooperation and competition.

INTRODUCTION

We focuse our attention on the implementation of simulations where automatic organizations appear in function of interaction over some systems. Many evolutive and complex systems in natural and artificial worlds produce such dynamical organizations which born, grow and die due to the interactions with each others and with their environment.

Agent-based simulations are constructed on decentralized approaches for computational implementations. The description of their entities in term of autonomy and communication gives to these models interesting properties for decentralized problems solvers and for computable representation of complex systems.

The model of individual agent behaviour is essential for the representation of its interaction with the system and its autonomy characteristic. A behaviour description based on operating model allows to manage automatic treatments. In this paper, we use automata-based model for the agent behaviour description. Many classical operators are well-defined and allow us to implement automatic treatments. Dynamical and adaptive properties can be described in term of specific operators based on genetic algorithms.

In the following, we first present the bases of our models in term of automata with multiplicities and especialy in term of probabilistic automata. Then, we present the dilemma prisoner as a general model for competition and cooperation models in distributed modelisation. So, we defined the genetic operators on probabilistic automata which allow to model adaptive behaviour for prisoner dilemma strategies. Some experimentations of such adaptive strategies are then shown and conclude this paper.

AUTOMATA WITH MULTIPLICITIES FOR AGENT BEHAVIOUR MODEL

In multi-agent systems, the agent behaviour can be implemented with automata, well-suited for the representation of different comportemental states and for the mecanisms representation of the transitions beetween these states.

Thus, automata with multiplicities (Schützenberger, 1961), defined below, are automata with outputs which belong to a semiring. These automata are used in the studies described in this paper because of the great numbers of operations which can be defined on them, classical ones (Duchamp et al., 1999) as well as evolutive agent oriented ones (Bertelle et al., 2001).

An automaton with multiplicities over a finite alphabet Σ and a semiring $(K, \oplus, \odot)$ is a $5-tuple$ $(\Sigma, Q, I, T, \delta)$, with Q a finite set of states and I, T, δ being mappings such that $I : Q \to K$, $T : Q \to K$, and $\delta : Q \times \Sigma \times Q \to K$. I

Proceedings 14th European Simulation Symposium
A. Verbraeck, W. Krug, eds. (c) SCS Europe BVBA, 2002

(resp. T) defines initial states (resp. final states) and δ is the transition function.

When the set of elementary outputs is the alphabet of a semiring (for example Π^*, for Π an alphabet), the automaton is a transducer. So, transducers are suited to model agent behaviours: the input alphabet corresponds to the agent perceptions and the output alphabet corresponds to the agent actions.

The framework of automata with multiplicities also allows us to model probabilistic aspects in agent behaviour: each internal transition produces in output the probability of its realization. Indeed, a stochastic (or probabilistic) automaton is a particular case of multiplicities one (where $K = [0, 1]$). In comparison with the definition given above, the transition function δ defined above becomes $\delta : Q \times \Sigma \times Q \to [0, 1]$, with the constraint:

$$\forall a \in \Sigma, \forall q \in Q \quad \sum_{p \in Q} \delta(q, a, p) = 1 \,. \tag{1}$$

The sum of probabilities of all transitions for each perception, starting from any state, must be equal to 1.

In order to realize operations on automata, we use a linear representation of automata. We describe this one below, especialy for probabilistic automata. Such a representation of dimension n (number of states) is a triplet (λ, μ, γ) where:

- $\lambda \in [0, 1]^{1 \times n}$, with $\sum_{i=1}^{n} \lambda_{1,i} = 1$, a row vector coding the input probabilities,
- $\gamma \in \{0, 1\}^{n \times 1}$ a column vector coding the output probabilities,
- $\mu : \Sigma^* \to [0, 1]^{n \times n}$ a morphism of monoids coding the transition probabilities between states for each letter $a \in \Sigma$, with the matrix $\mu(a)$.

The constraint represented by expression (1) is equivalent to

$$\forall a \in \Sigma, \forall q \in Q \quad \sum_{p \in Q} \mu_{q,p}(a) = 1 \,. \tag{2}$$

A successful path in an automaton is a path from an initial state to a final one. Then, for all $w \in \Sigma^*$ corresponding to a successful path, $\lambda\mu(w)\gamma$ is the probability of all successful paths labelled by w (it is the probability that the successful succession of perceptions, w, occurs).

In the following, we show illustrative examples of such agent model and we give the linear representation for them.

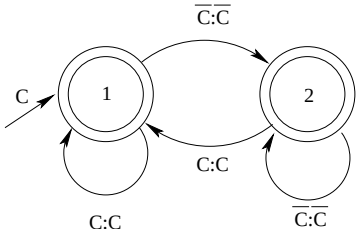
PRISONER DILEMMA : AUTOMATA BASED MODEL FOR COOPERATION AND COMPETITION ASPECTS

In multi-agent systems, interactions lead to social behaviour. They are often described in terms of cooperation or competition aspects. We focuse our attention with prisoner dilemma (Axerold, 1984) which is a model for negociation behaviour, allowing alternance of cooperation and competition between agents of the same system.

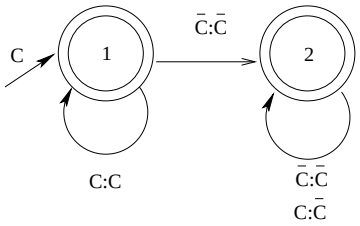
The prisoner dilemma is a two-players game where each player has two possible actions: cooperate (C) with its opponent or betray it ($\overline{C}$). So, four outputs are possible for the global actions of the two players. A relative payoff is defined relatively to these possible outputs, as described in the following table where the rows correspond to one player behaviour and the columns to the other player one. For example, if the player one betrays the other which tries to collaborate with him, the payoff for the first player is 5 and for the other 0 (5,0).

	C	$\overline{C}$
C	(3,3)	(0,5)
$\overline{C}$	(5,0)	(1,1)

Table 1: Prisoner dilemma payoff



Strategy A : Tit-for-tat strategy



Strategy B : Vindictive strategy

Figure 1: Two prisoner dilemma strategies in term of tranducers

In iterative version of the prisoner dilemma, successive steps can be defined. Each player don't know the action of its opponent during the current step but he knows it for the precedent step. So different strategies can be defined for player behaviour, the goal of each one is to obtain maximal payoff for himself. In the figure 1, we describe two strategies with transducers. Each transition is labeled by the input corresponding to the player perception which is the precedent opponent action and the output corresponding to the present player action. The only inital state is the state 1,

recognizable by the incomming arrow labeled only by the output. The final states are the states 1 and 2, recognizable with the double circles. In strategy A, the player has systematically the same behaviour as his opponent at the previous step. This strategy very simple is known as one of the best. In strategy B, the player chooses definitively to betray as soon as his opponent does it once.

These previous automata represent static strategies and so they are not well suited for the modelization of evolutive strategies. For this purpose, we propose a model based on a probabilistic automaton (see fig. 2).

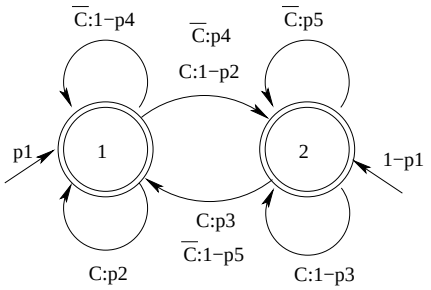


Figure 2: Probabilistic multi-strategies two-states automata for prisoner dilemma

This automaton represents all the two-states strategies for cooperation (C) and competitive ($\overline{C}$) behaviour of one agent against another in prisoner dilemma. The transitions are labeled in output by the probabilities (p_i) of their realization. The state 1 is the state reached after cooperation action and the state 2 is reached after betrayal.

For this automaton, the associated linear representation, as described previously, is:

$$\lambda = (p_1, 1-p_1) \qquad \gamma^t = (1,1)$$

$$\mu(C) = \begin{bmatrix} p_2 & 1-p_2 \\ p_3 & 1-p_3 \end{bmatrix}$$

$$\mu(\overline{C}) = \begin{bmatrix} 1-p_4 & p_4 \\ 1-p_5 & p_5 \end{bmatrix}$$

In the following section, we describe a general genetic algorithm on probabilistic automata. We show how it can be applied for modelling an adaptive strategy for the prisoner dilemma based on the previous particular probabilistic automaton.

GENETIC ALGORITHMS ON PROBABILISTIC AUTOMATA

We describe a genetic algorithm managing a population, that is the agent behaviours coded with probabilistic automata. Genetic algorithms use individual characteristic representations, named chromosomes. We define the chromosome for each agent as the sequence of all the matrices $\mu(a)$ associated to each perception $a \in \Sigma$. In the following, genetic algorithms will generate new automata containing eventually new transitions from the ones included in the initial automata. To authorize only significant behaviours, we have to consider the existence of a family of boolean transition matrices $(\mathcal{T}_a)_{a\in\Sigma}$, associated to each type of agent, and coding all its possible transitions for each perception. The effective transitions matrix associated to each perception is a "subset" of it (in fact, each transition matrix associated to a given perception $a \in \Sigma$, denoted $\mu(a)$, is a matrix of same dimension as $\mathcal{T}_a$, but with probabilistic coefficients).

In the genetic algorithm, each couple of agents follows a reproduction iteration broken up into three steps:

- Duplication where each agent of the couple generates a clone of itself;
- Crossing-over where a sequence of lines of each matrix $\mu(a)$ for all $a \in \Sigma$ is arbitrary chosen. For each of these matrices, a permutation on the lines of the chosen sequence is made between the respective matrices of the two agents corresponding to the reproduction couple. In term of automata operator, the crossing over is the permutation of all the transitions outgoing from all states/lines selected by the crossed operation. This permutation respects the expression (2).
- Mutation where a line for each matrix $\mu(a)$ is arbitrary chosen and, randomly, a sequence of new values is affected to this line, according to the probabilistic nature of the matrix represented by the expression (2). The new matrix obtained by mutation must respect the authorized transitions given by the $(\mathcal{T}_a)_{a\in\Sigma}$ family.

The reproduction steps generate new agents behaviours. Genetic algorithms have to select among these behaviours, some of ones which have the best values of a given function called fitness. In dilemma prisoner, the fitness function returns, for a given behaviour automaton, the corresponding payoff value.

Finally, the whole genetic algorithm scheduling for a full process of reproduction over all the agents is the evolutionary algorithm:

1. For every couple of agents, two children are created by duplication, crossing over and mutation mechanisms;
2. The fitness, for every agent, is computed;
3. For every 4-tuple composed of parents and children, the two performless agents, in term of fitness computed in previous step, are removed. The two agents, still alive, result of the evolution.

EVOLUTIVE ADAPTATION FOR PRISONER DILEMMA: IMPLEMENTATION AND SIMULATION RESULTS

Genetic solvers have yet been experimented on the iterative prisoner dilemma. In (Holland, 1995), the chromosomes of the algorithm is a 64 positions string corresponding to each player memory of the past three outputs. We propose a more generic approach using genetic algorithms for prisoner dilemma as a particular application of the genetic algorithm on general probabilistic automata, as described in previous section. So, this algorithm is applied to the particular probabilistic automaton described in figure 2. This allows to simulate adaptive behaviours in term of evolutive strategies. Implementations have been made (Frebourg, 2001), using the Madkit platform (Gutknecht and Ferber, 1997) developped in LIRMM (Montpellier - France). It shows adaptive strategies improving the player payoff.

A first sequence of experimentations has been made. These experimentations consist in building on one hand, an adaptive agents population described by the probabilistic automaton of the figure 2 and on the other hand, some agents with static behaviour. The opponent players in prisoner dilemma are built from these two kinds of agents. Genetic algorithms are used on the adaptive agents population which fitness is the player payoff. Against the static tit-for-tat strategy, the adaptive behaviour converges, after 250 iterations, to the strategy described by the linear representation:

$$\mu(C) = \begin{bmatrix} 0.99637 & 0.00363 \\ 0.99259 & 0.00741 \end{bmatrix}$$

$$\mu(\overline{C}) = \begin{bmatrix} 0.88216 & 0.11784 \\ 0.98521 & 0.01479 \end{bmatrix}$$

So, the emerging strategy seems to be the one which consists in cooperating whatever the perception. The agent payoff average associated to this emerging strategy is 3.

Another sequence of experimentations has been made. Two agents populations are opposed. The first one is composed of 12 static strategies. The other one is composed of adaptive agents which evolve genetically to a strategy which must be efficient against many ones. The first result obtained is an effective convergence, after 250 iterations, to the strategy described by the linear representation:

$$\mu(C) = \begin{bmatrix} 0.08589 & 0.91411 \\ 0.93810 & 0.06190 \end{bmatrix}$$

$$\mu(\overline{C}) = \begin{bmatrix} 0.75001 & 0.24998 \\ 0.37071 & 0.62929 \end{bmatrix}$$

The agent payoff average is here 2.4.

These first results show that the adaptive strategy modelization gives a convergent process able to improve the payoff. Some parameters have to be fixed like the mutation rate whose value has been fixed to 0.5% in the previous experimentations. Complementary experiments have to be studied to confirm the efficiency of the method.

CONCLUSION

The study presented in this paper is based on the use of automata with multiplicities for agent behaviour modelling. The great number of operators defined on these automata allows us to implement automatic process on agent behaviours. In transducer based formulation, the input and ouput alphabets give a representation of agent perceptions and actions. In probabilistic based formulation, multi-strategies behaviours can be model and genetic algorithms defined on them allow to generate adaptive behaviours. Adaptive multi-strategies have been modeled with such automata for the prisoner dilemma and efficient strategies can be automatically found with genetic algorithms. We project to use this elementary multi-strategies adaptive automaton as a generic model for agents interactions which are able to evolve dynamically from cooperation to competition or vice versa.

REFERENCES

Axerold, R. (1984). *The evolution of cooperation*. New York Basic Book.

Bertelle, C., Flouret, M., Jay, V., Olivier, D., and Ponty, J.-L. (2001). Genetic algorithms on automata with multiplicities for adaptive agent behaviour in emergent organizations. In *SCI'2001*, Orlando (USA).

Duchamp, G., Flouret, M., and Laugerotte, E. (1999). Operations over automata with multiplicities. In Champarnaud, J.-M., Maurel, D., and Ziadi, D., editors, *WIA'98*, volume 1660 of *Lecture Notes in Computer Science*, pages 183–191. Springer-Verlag.

Frebourg, V. (2001). Mise en œuvre de la représentation du comportement d'un agent. Technical report, LIH, Le Havre University.

Gutknecht, O. and Ferber, J. (1997). Madkit: Organizing heterogeneity with groups in a platform for multiple multi-agents systems. Technical report, LIRMM, Montpellier University, http://www.madkit.org.

Holland, J. H. (1995). *Hidden Order - How adaptation builds complexity*. Perseus Book.

Schützenberger, M. (1961). On the definition of a family of automata. *Inform. and Control*, 4:245–270.