Matis MIS7
Chapitre 4 : Agents, automates génétiques, applications et implémentations

Cyrille Bertelle

LITIS
Laboratoire d'informatique,
du Traitement de l'Information et des Systèmes
Université du Havre

Master 2 Math-Info - Matis

Plan

1. Systèmes Multi-Agents
2. Modèle de comportement d'agents par des automates à multiplicités
3. Application à la théorie des jeux
4. Automates génétiques spatialisés
5. Eco-résolution : modélisation décentralisée de résolution de problèmes

Systèmes Multi-Agents

Agent concept

- Agere (latin): to do
- Def. 1: An agent is a computational entity such as a software or a robot
  - that can be viewed as **perceiving and acting** upon its **environment**,
  - that is **autonomous** (it has its own life cycle and can act from some internal reasoning)
- Def. 2: An Agent is a system that decides for itself what it needs to do in order to satisfy its **objectives** or **goals.**

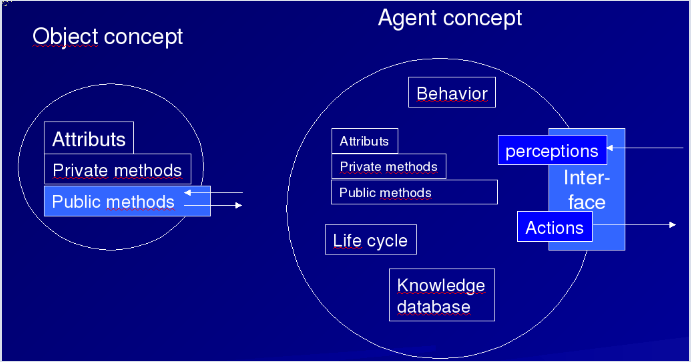
Systèmes Multi-Agents

Agent-based programming as an evolution from object-oriented programming

- From sequential/procedural programming to object-oriented programming
- From object-oriented programming to agent-based programming

Systèmes Multi-Agents

From object-oriented to agent-based programming



Systèmes Multi-Agents

Why this evolution?

Evolution from artificial intelligence to Distributed Artificial Intelligence (DAI) associated to Multiagent Systems

- Development of complex systems concept: new way of modelling based on emergence from interacting entities
  $\rightarrow$ We need efficient way of programming for that purpose
- Development of distributed computing based on huge computer networks

Systèmes Multi-Agents

Intelligent agent classification based on behavioral model

- The agent is **reactive**: from a given perception, he always do same simple action.
- The agent is **cognitive**: the agent can use some advances computation based on knowledge data-base, learning, life-cycle, ...

Systèmes Multi-Agents

Agent: some elements of implementation

- Agent can be seen as object extension: Autonomy+communication system
- Autonomy can be represented with life-cycle (initiated, making jobs and killed)
  $\longrightarrow$ Each agent is in a thread (light autonomous program which share memories with other threads)

Systèmes Multi-Agents

MultiAgent Systems

- A set of interacting agents
- which are using communications between them, as interacting processes
- and able to be involved in organizations (predefined organizations or self-organizations)

Systèmes Multi-Agents

What we can do with ABSM that others models make with difficulties

- Dealing with multi-scale models and scales transfer:
  - fine-scale local phenomena description
  - link and make interact these fine-scale phenomena with high-level descriptions (socioeconomic equations, ...)
- Dealing with model complex dynamical systems:
  - links between environmental, economical and social aspects with human decision

Systèmes Multi-Agents

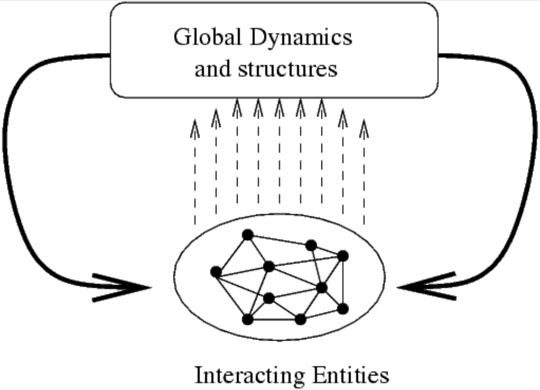
What we can do with ABSM that others models make with difficulties

- Individual-based modeling and heterogeneity
- Modeling interaction processes in detail by interaction networks for socioeconomic and environmental processes
- Then using emergent computing over these interaction networks to detect self-organization

Plan

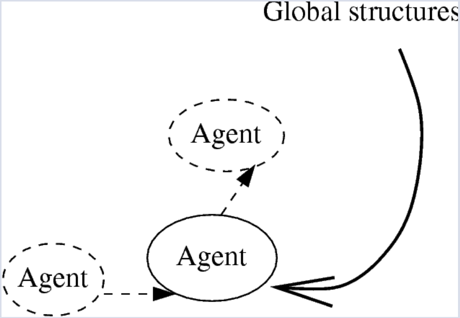
Automates à multiplicités

Multi-agent for complex systems



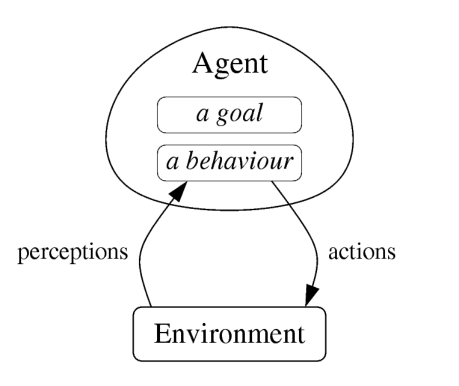
Automates à multiplicités

Interactive Agents System



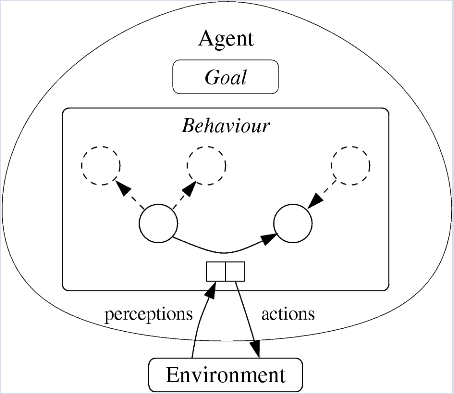
Automates à multiplicités

Agents modeling



Automates à multiplicités

Automata-based Agent Behavior



Automates à multiplicités

Automata with multiplicities

Agent behavior is modelled by automata with multiplicities which is defined by

- A set of perception represented by an alphabet
- A set of actions represented by a **semi-ring** K
- A set of states with a subset of initial states and a subset of final states
- A set of transitions between states which is generate by a perception (input) and leads to an action according to the probabilities

Automates à multiplicités

Automata with multiplicities

Because the set of actions K is a **semi-ring**,

- we can represented the automata using a linear representation (vectors and matrices),
- we can defined many kinds of operators on these automata and so improve automatic processes on agent management

Automates à multiplicités

Applications

Many applications can be modeled by Agent-based systems modelling:

- Decision Support Systems
- Game theory

Plan

Application à la théorie des jeux

Prisonner Dilemma

- Two players game involving cooperation/competition model
- Basic model for economic purpose
- Rules:
  - 2 options for each play: cooperate (C) or defect (D)
  - associated payoff for each situation in following table

Application à la théorie des jeux

Prisonner Dilemma Payoff Table

		Player 2	
		Cooperate (C)	**Defect (D)**
Player 1	**Cooperate (C)**	(+3, +3)	(0, +5)
	Defect (D)	(+5, 0)	(+1, +1)

Application à la théorie des jeux

Prisonner dilemma application

- Great number of applications, especially in strategical or economic domains
- Example: competition between two companies.
  - aggressive competition behavior
  - cooperative behavior
  - ... see payoff table in the following slide

Application à la théorie des jeux

Prisonner Dilemma Application Payoff Table

	Company C1		
Company C2		**Cooperative politic**	**Aggressive politic**
	Cooperative politic	Medium profit for each	Poor profit for C2 and huge profit for C1
	Agressive politic	Poor profir for C1 and huge profit for C2	Poor profit

Application à la théorie des jeux

Iterative version for prisonner dilemma

- Successive steps
- Each player does not know adversary action ...
- ... but he knows the previous action of his adversary
- So different strategies can be defined for player behavior (goal: having the maximal payoff for himself)

Application à la théorie des jeux

Some strategies

- Vindicative strategy:
  - If the adversay cooperates at the previous play, I cooperate;
  - If the adversary defects one time, I will always defect whatever the adversary will do latter.
- Tit-for-tat strategy:
  - I always do what my adversary had to do at the previous play

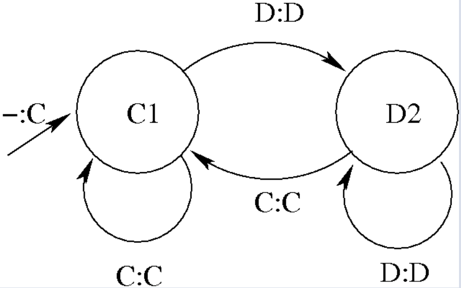
Application à la théorie des jeux

Tit-for-tat strategy model

- Two states: "C1" means "I cooperate", "D2" means "I defect"
- Transition for each play:
  - Input: what the adversary do at the previous play
  - Output: what I do in consequence
- Tit-for-tat strategy: I always do what the adversory do at the previous play

Application à la théorie des jeux

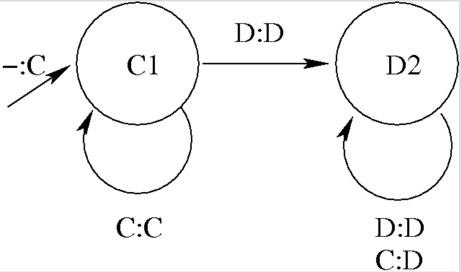
Tit-for-tat strategy model



Application à la théorie des jeux

Vindictive strategy model

Another example: in this strategy, once the adversory defect, we always defect after.



Application à la théorie des jeux

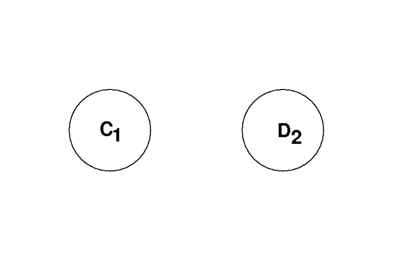
Multi-strategy model, using probabilistic automaton

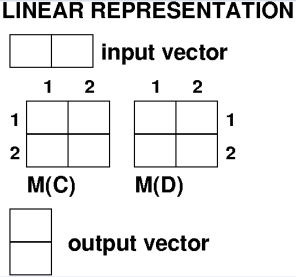
Probabilistic transition from one state to another according to what make the adversory at the previous step

Application à la théorie des jeux

Multi-strategy model, using probabilistic automaton

Probabilistic transition from one state to another according to what make the adversory at the previous step

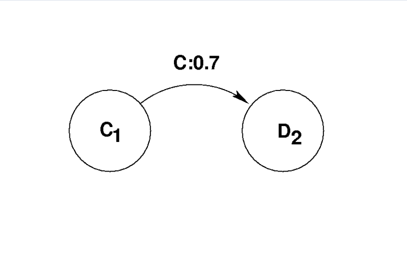


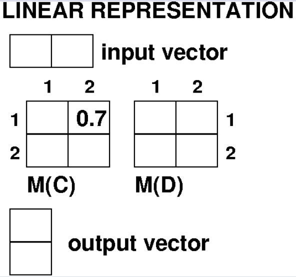


Application à la théorie des jeux

Multi-strategy model, using probabilistic automaton

Probabilistic transition from one state to another according to what make the adversory at the previous step

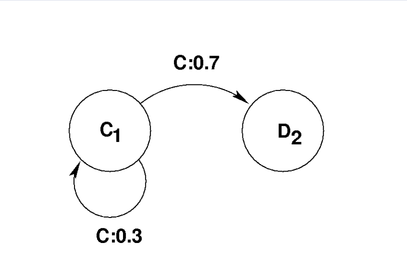


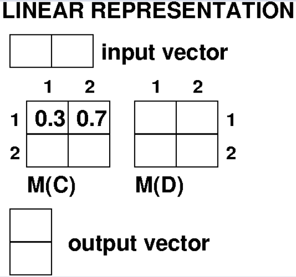


Application à la théorie des jeux

Multi-strategy model, using probabilistic automaton

Probabilistic transition from one state to another according to what make the adversory at the previous step

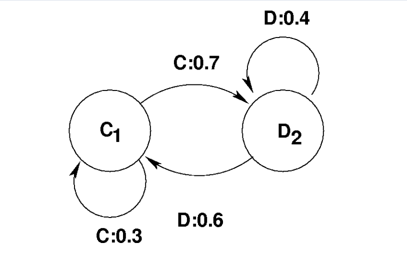


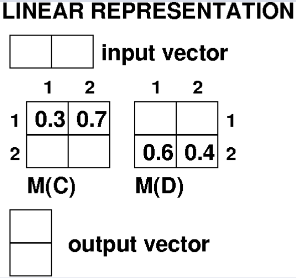


Application à la théorie des jeux

Multi-strategy model, using probabilistic automaton

Probabilistic transition from one state to another according to what make the adversory at the previous step

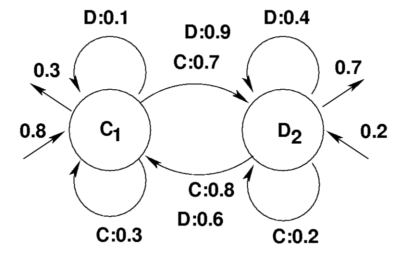




Application à la théorie des jeux

Multi-strategy model, using probabilistic automaton

Probabilistic transition from one state to another according to what make the adversory at the previous step



LINEAR REPRESENTATION

0.8	0.2

input vector

M(C):

	1	2
1	0.3	0.7
2	0.8	0.2

M(D):

1	2	
0.1	0.9	1
0.6	0.4	2

0.3
0.7

output vector

Plan

Automates spatialisés de comportements

Spatial automata and associated distances

- A spatial agent is defined by
  - Spatial coordinates
  - A behavior modeled by an automaton with multiplicities
- A **spatial distance** between 2 agents, can be computed according to their spatial coordinates
- A **behavioral distance** between 2 agents can be compute by the distance between the vectors which stores all the coefficients of the linear representation of the agent behavior automata.

Automates génétiques

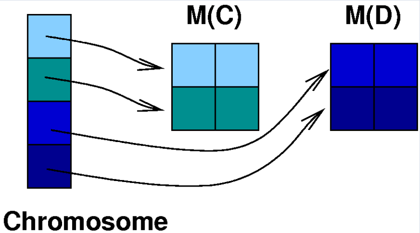
Genetic operators on automata population

- Genetic operators deal with **population** of **individuals** as spatial behavioral automata.
- **Individual** is described by a **chromosome** which is a sequence of **alleles** (elementary information).
- Here, the chromosomes are coding the transition matrices of the behavioral automata linear representation.
- Here, an allele is a matrix line ...

Automates génétiques

Genetic operators on automata population

- ... and the chromosome is the set of the matrix lines of all the transition matrices



Community Swarm Optimization Algorithm

Overview

- CSO Algorithm consists in generating an initial virtual automata population describing some spatial transition rules system;
- This virtual automata population evolves and moves on a spatial environment;
- The evolution (and the moving) follows a genetic algorithm including a selection process associated to a fitness function.

Community Swarm Optimization Algorithm

Community Detection associated to fitness function

- We can defined the fitness of each agent as following:
  - We compute his neighbourhood, using the ***spatial distance***
  - We sum the ***behavioral distance*** of the agent itself with all the others agents included in the neighbourhood
  - We define the fitness as the inverse of the average of the previous sum.
- Self-organization communities emerge from the use of this fitness inside a genetic algorithm.

Community Swarm Optimization Algorithm

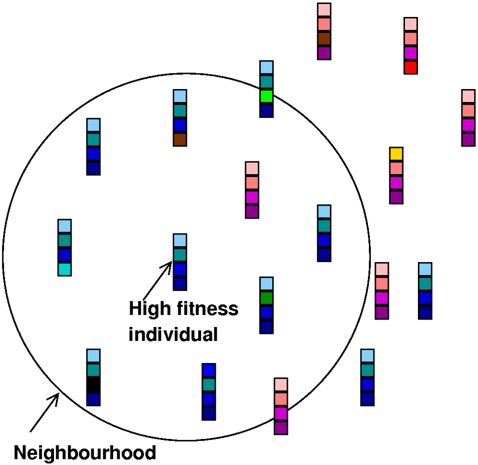
Community Detection associated to fitness function

Let $\mathcal{V}_x$ a neighbourhood of the agent x, relatively to its spatial location. We define $f(x)$ the agent fitness of the agent x as :

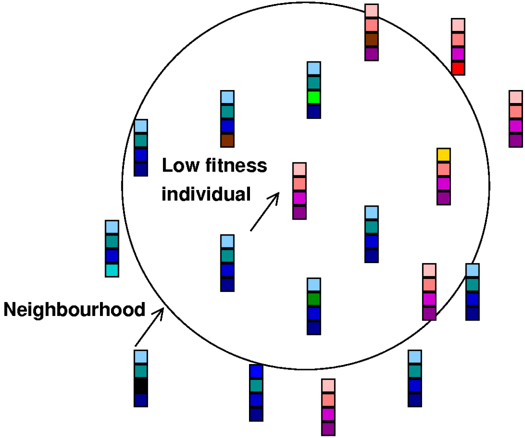
$$f(x) = \begin{cases} \dfrac{card(\mathcal{V}_x)}{\sum\limits_{y_i \in \mathcal{V}_x} d(x, y_i)^2} & \text{if } \sum\limits_{y_i \in \mathcal{V}_x} d(x, y_i)^2 \neq 0 \\ \infty & \text{otherwise} \end{cases}$$

where $d(x, y)$ is the behavioral semi-distance between the two agents x and y.

CSO Algorithm: example ... following



CSO Algorithm: example ... following



Community Swarm Optimization

Recherche en cours au LITIS

- Thèse de Rawan Ghnemat
- Un chapitre de livre en cours d'édition (Springer)
- Plusieurs publications/communications en revues et conférences

Plan

1. Systèmes Multi-Agents
2. Modèle de comportement d'agents par des automates à multiplicités
3. Application à la théorie des jeux
4. Automates génétiques spatialisés
5. Eco-résolution : modélisation décentralisée de résolution de problèmes

Eco-résolution : modélisation décentralisée de résolution de problèmes

Principes de l'éco-résolution

- Basé sur des agents réactifs qui intéragissent ;
- Approche décentralisée de résolution d'un problème par émergence du système d'interaction des agents ;
- On cherche un état stable considéré comme la solution du problème ;

Eco-résolution : modélisation décentralisée de résolution de problèmes

Les éco-agents

L'agent a un **but** : être satisfait.

L'agent et son environnement :

- Son environnement est constitué des autres agents voisins dans un réseau d'accointance ou de dépendances et avec lesquels il va interagir par perceptions/actions ;

Eco-résolution : modélisation décentralisée de résolution de problèmes

Les éco-agents

- Ses perceptions :
  - A : Il est agresser par un autre agent ;
  - G : Il est géné par d'autres agents.
- Ses actions :
  - FS : Faire satisfaction (réalisation de l'action l'amenant à son but) ;
  - Ag : Agresser un autre agent ;
  - FF : Faire fuite.

Eco-résolution : modélisation décentralisée de résolution de problèmes

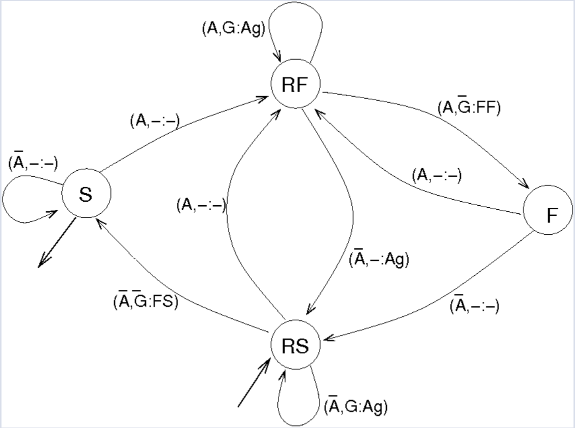
Architecture de l'éco-agent

Un éco-agent peut être représenté par un automate à états finis (automate à multiplicités) :

- Il a 4 états internes :
  - S : il est satisfait (état de sortie) ;
  - RS : il est en recherche de satisfaction (c'est arbitrairement l'état d'entrée) ;
  - RF : il est en recherche de fuite ;
  - F : il est en état de fuite.
- On définit un système de transitions à sorties (multiplicités) entre ces états (cf. la figure suivante)

Eco-résolution : modélisation décentralisée de résolution de problèmes

Automate du comportement d'un éco-agent



Eco-résolution : modélisation décentralisée de résolution de problèmes

Action de l'agent dans son réseau d'accointance/dépendances

- Quand un agent se satisfait, il informe ses dépendances qu'elles peuvent se satisfaire ;
- Quand un agent ne peut se satisfaire, il recherche les géneurs parmi ses accointances et les agresse ;
- Quand un agent cherche à fuir, il recherche les géneurs parmi les accointances et les agresse.

Eco-résolution : modélisation décentralisée de résolution de problèmes

Références et applications

- J. Ferber *“Les systèmes multi-agents”*, InterEditions, 1995.
- A. Drogoul et C. Dubreuil *“Eco-Problem Solving: results of the N-Puzzle”* in Decentralized Artificial Intelligence 3, Y. Demazeau et E. Werner (eds), North Holland, 1992.
- Applications :
  - Problème d’allocations de tâches (Agents tâches et agents ressources) ;
  - Problème de chaînes de production ;
  - Simulation de la stabilité de structures tourbillonnaires émergentes dans des écoulements de fluides.

... compléter par des recherches personnelles sur ces références et applications.