A selection of MAS learning techniques based on RL

Ann Nowé

Vrije Universiteit Brussel

Content

Single stage setting

- Common interest (Claus & Boutilier, Kapetanakis&Kudenko)
- Conflicting interest (Based on LA)

Key questions

Are RL algorithms guaranteed to converge in MAS settings?
If so, do they converge to (optimal) equilibria?

Are there differences between agents that learn as if there are no other agents (i.e. use single agents RL algorithms) and agents that attempt to learn both the values of specific joint actions and the strategies employed by other agents?

How are rates of convergence and limit points influenced by the system structure and action selection strategies?

Simple single stage

common deterministic interest game

	a_0	a_1
b_0	x	0
b_1	0	y

If x > y > 0, (*a0, b0*) and (*a1, b1*) 2 equilibria
first one is optimal
If x = y > 0 equilibrium selection problem

Super RL agent (Q-values for joint actions and joint action selection)
No challenge, equivalent to single agent learning

Joint action learners (Q-values for joint actions, actions are selected independently)

Independent learners (Q-values for individual actions, actions are selected independently)

Simple single stage

common deterministic interest game

Joint action learners (Q-values for joint actions, actions are selected independently)

Use e.g. Q-learning to learn Q(a_0, b_0), Q(a_0, b_1) , Q(a_1, b_0) and Q(a_1, b_1)

Assumption: actions taken by the other agents can be observed.

Action selection for individual agents:

the quality of an individual action depends on the action taken by the other agent-> maintain beliefs about strategies of other agents.

$$EV(a^i) = \sum_{a^{-i} \in A_{-i}} Q(a^{-i} \cup \{a^i\}) \prod_{j \neq i} \{\Pr^i_{a^{-i}[j]}\}$$

Simple single stage

common deterministic interest game

Independent learners (Q-values for joint action, actions are selected independently

- Use e.g. Q-learning to learn Q(a_0), Q(a_1), Q(b_0) and Q(b_1)
  - No need to observe actions taken by other agents.
- Action selection for individual agents:
  - Exploration strategy is crucial
  - (Random not OK, Boltzmann with decreasing T is Ok)

Simple single stage

Comparing Independent Learners and Joint action learners

	a_0	a_1
b_0	10	0
b_1	0	10

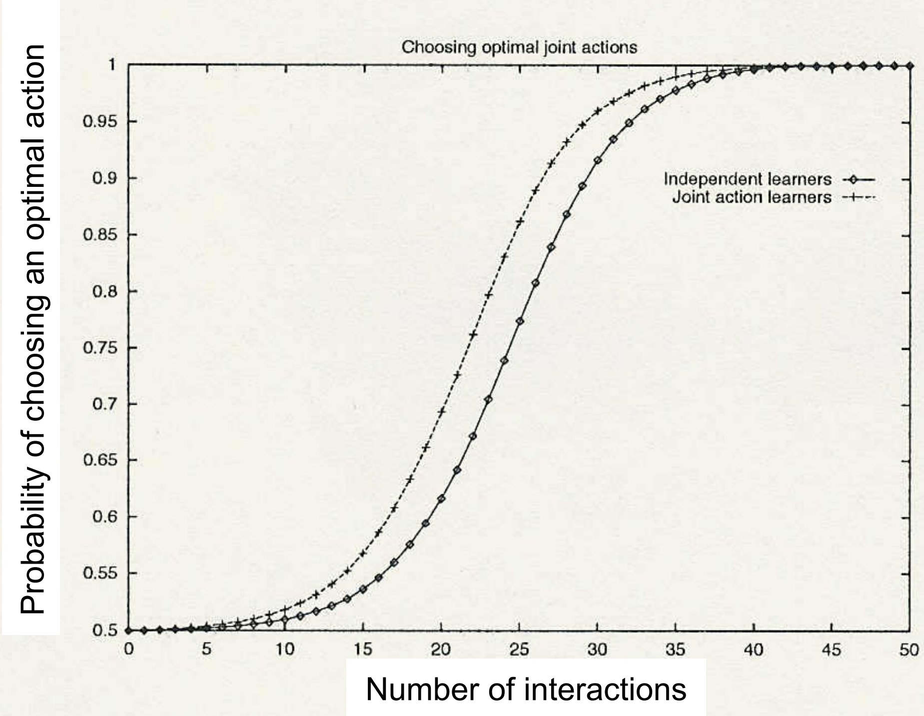


Figure 1: Convergence of coordination for ILs and JALs (averaged over 100 trials).

The penalty game

	a_0	a_1	a_2
b_0	10	0	k
b_1	0	2	0
b_2	k	0	10

$k < 0$

3 Nash Equilibria , 2 optimal

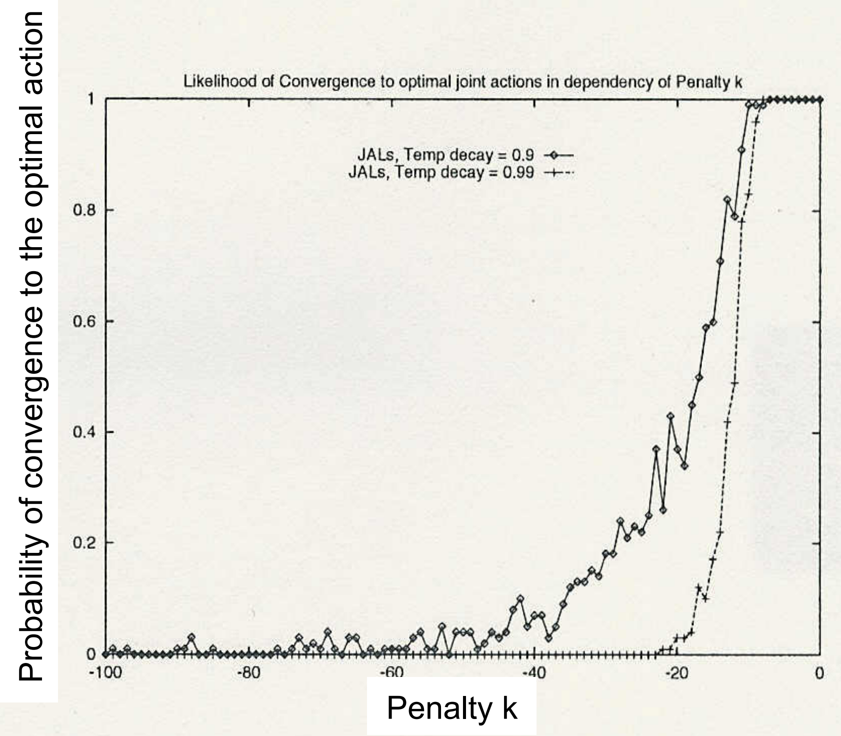


Figure 2: Likelihood of convergence to opt. equilibrium as a function of penalty k (averaged over 100 trials).

Similar results hold for IL with decreasing exploration

Climbing game

	a_0	a_1	a_2
b_0	11	-30	0
b_1	-30	7	6
b_2	0	0	5

2 Nash Equilibria , 1 optimal

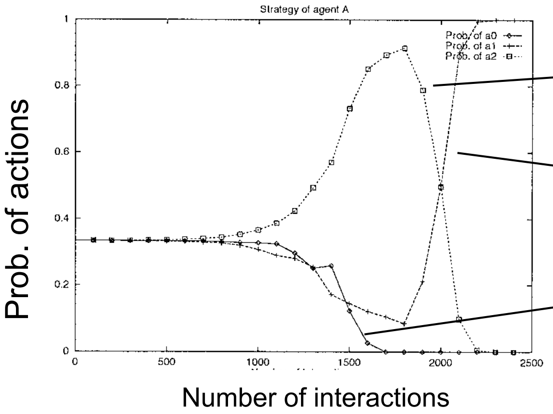


Figure 3: A's strategy in climbing game

Action a2

Action a1

Action a0

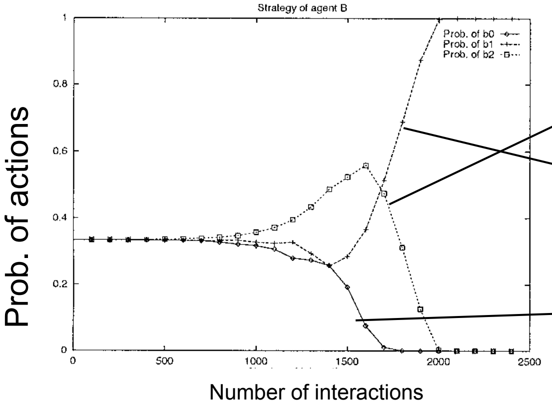


Figure 4: B's strategy in climbing game

Action b2

Action b1

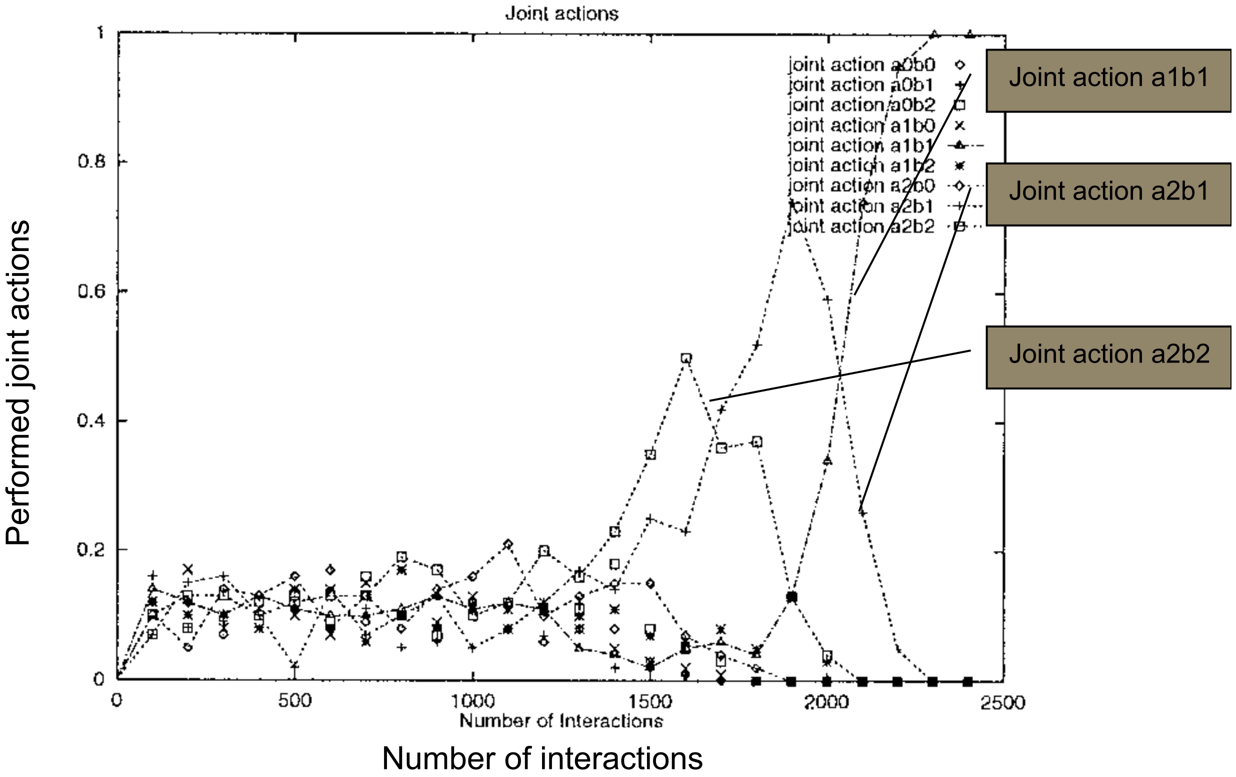
Action b0

initial temperature 10000 is decayed at rate 0.995

Climbing game

	a_0	a_1	a_2
b_0	11	-30	0
b_1	-30	7	6
b_2	0	0	5

2 Nash Equilibria ,
1 optimal



Joint action a1b1

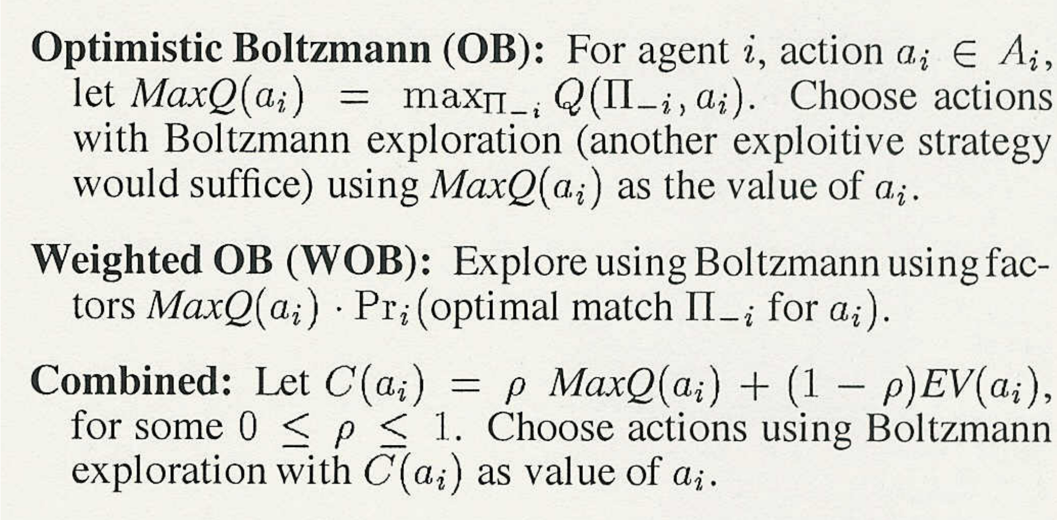
Joint action a2b1

Joint action a2b2

Figure 5: Joint actions in climbing game

initial temperature 10000 is decayed at rate 0.995

Biasing Exploration



Optimistic Boltzmann (OB): For agent i, action $a_i \in A_i$, let $MaxQ(a_i) = \max_{\Pi_{-i}} Q(\Pi_{-i}, a_i)$. Choose actions with Boltzmann exploration (another exploitive strategy would suffice) using $MaxQ(a_i)$ as the value of a_i.

Weighted OB (WOB): Explore using Boltzmann using factors $MaxQ(a_i) \cdot \Pr_i(\text{optimal match } \Pi_{-i} \text{ for } a_i)$.

Combined: Let $C(a_i) = \rho\ MaxQ(a_i) + (1 - \rho)EV(a_i)$, for some $0 \leq \rho \leq 1$. Choose actions using Boltzmann exploration with $C(a_i)$ as value of a_i.

	a_0	a_1	a_2
b_0	10	0	k
b_1	0	2	0
b_2	k	0	10

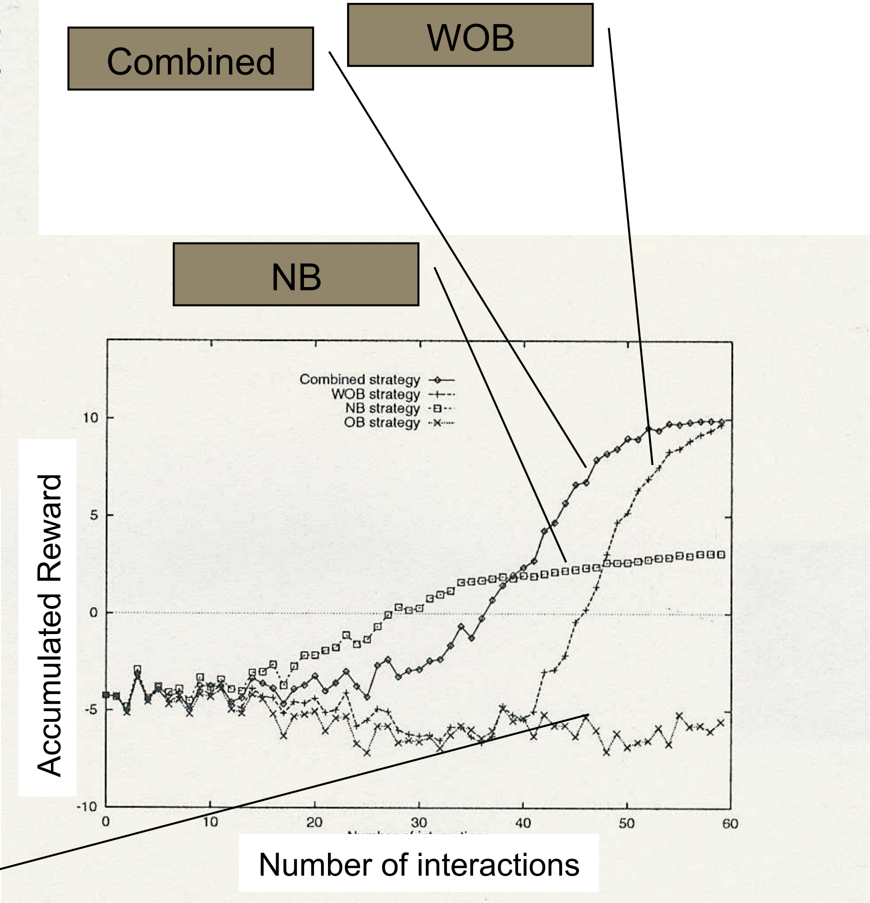


Figure 6: Sliding avg. reward in the penalty game

Content

Single stage setting

- Common interest (Claus & Boutilier, Kapetanakis&Kudenko)
- Conflicting interest (Based on LA)

FMQ Heuristic (Kapetanakis & Kudenko)

Observation:
The setting of the temperature in the Boltzmann strategy for independent learners is crucial.
Converge to some equilibrium, but not necessarily the optimal.
FMQ : Frequency Maximum Q value heuristic

$$EV(a) = Q(a) + c \times freq(\max R(a)) \times \max R(a)$$

Controls weight of heuristic

Fraction of time maxR(*a*)

Max reward so far for action *a*

$$p(a) = \frac{e^{\frac{EV(a)}{T}}}{\sum_{action' \in A_i} e^{\frac{EV(action')}{T}}}$$

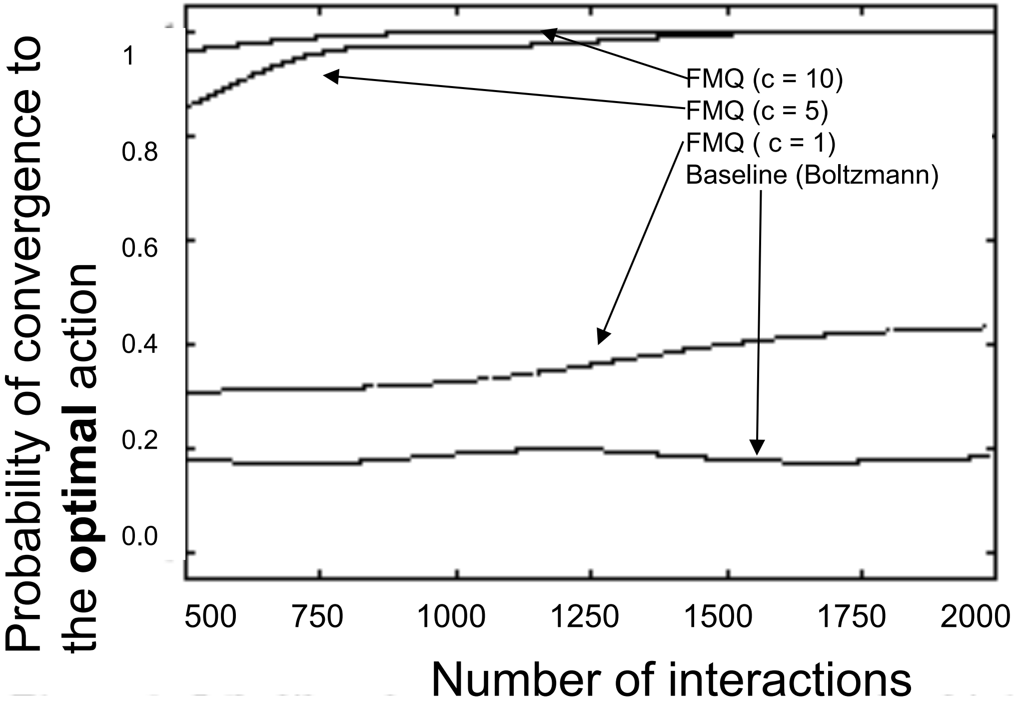
$$T(x) = e^{-sx} \times \max\_temp + 1$$

x number of iterations
s decay parameter

FMQ Heuristic (Kapetanakis & Kudenko)

	a_0	a_1	a_2
b_0	11	-30	0
b_1	-30	7	6
b_2	0	0	5

The climbing game



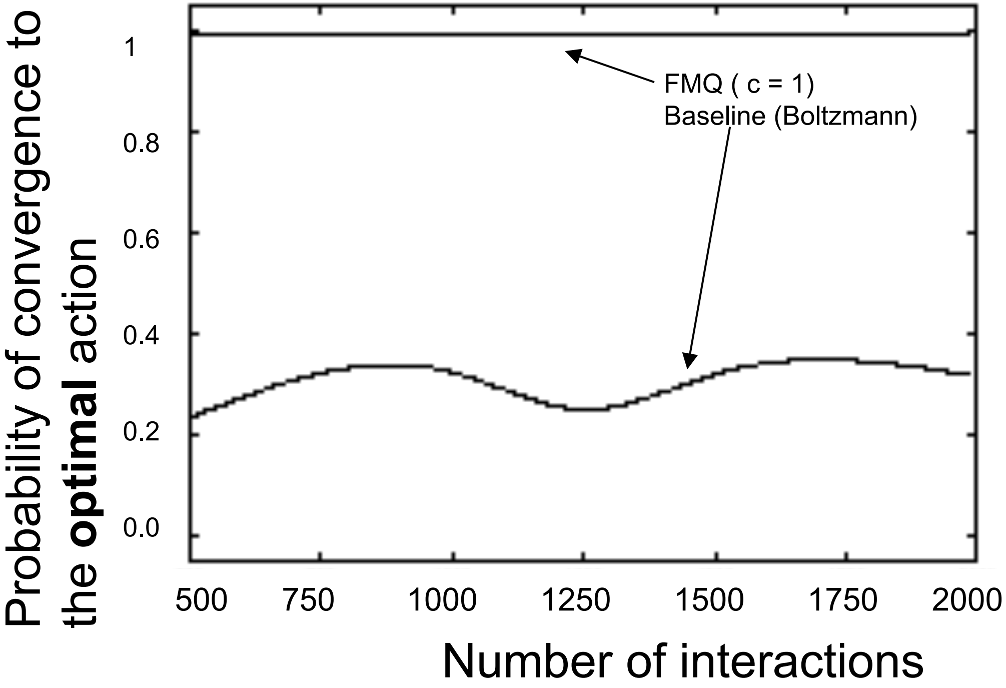
Likelihood of convergence to the optimal joint action (average over 1000 trials)

FMQ Heuristic (Kapetanakis & Kudenko)

	a_0	a_1	a_2
b_0	10	0	k
b_1	0	2	0
b_2	k	0	10

$k < 0$

The penalty game



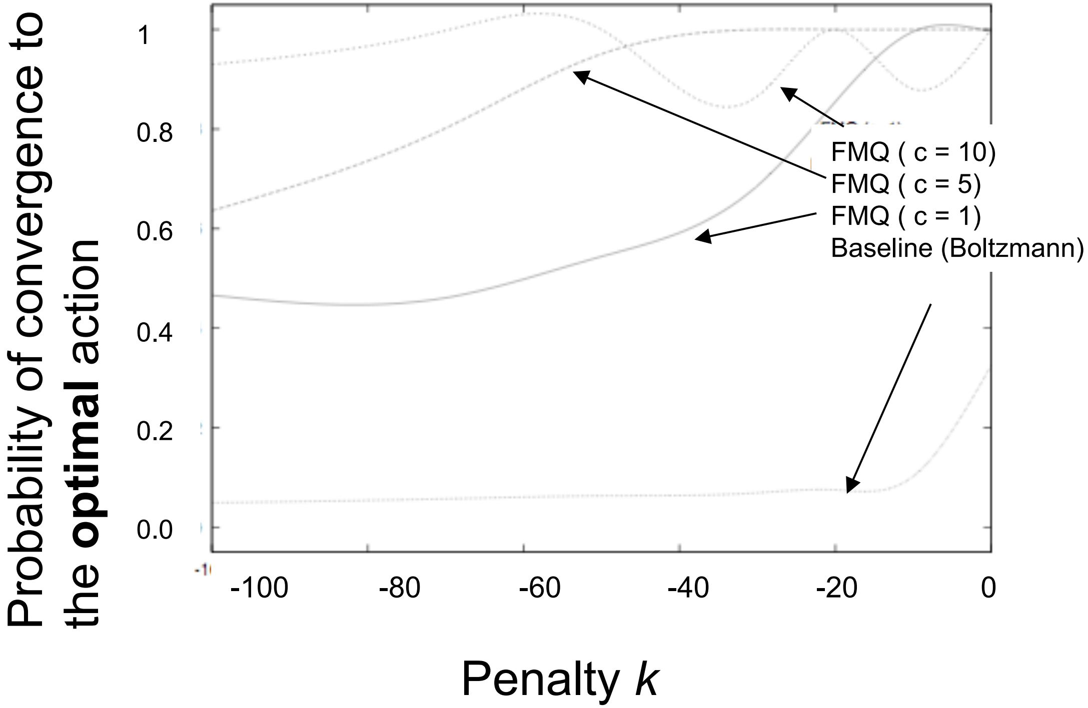
Likelihood of convergence to the optimal joint action (average over 1000 trials), k = 0

FMQ Heuristic (Kapetanakis & Kudenko)

	a_0	a_1	a_2
b_0	10	0	k
b_1	0	2	0
b_2	k	0	10

$k < 0$

The penalty game



Likelihood of convergence to the optimal joint action (average over 1000 trials, in function of k)

FMQ Heuristic (Kapetanakis & Kudenko)

The FMQ Heuristic is not very robust in stochastic reward games

	a_0	a_1	a_2
b_0	10/12	5/-65	8/-8
b_1	5/-65	14/0	12/0
b_2	5/-5	5/-5	10/0

The stochastic climbing game (50%)

GOAL is stochastic

Improvement : commitment sequences

Commitment Sequences (Kapetanakis & Kudenko)

- motivation: difficult to distinguish between the two sources of uncertainty (other agents, multiple rewards)
- definition: a commitment sequence is some list of time slots for which an agent is committed to taking the same action
- condition: an exponentially increasing time interval between successive time slots

Sequence 1: (1,3,6,10,15,22, …)
Sequence 2: (2,5,9,14,20,28, …)
Sequence 3: (4, …)

assumptions:

1. common global clock
2. common protocol for defining commitment sequences

Content

Single stage setting

- Common interest (Claus & Boutilier, Kapetanakis&Kudenko)
- Conflicting interest (Based on LA)

Learning Automata

Basic Definition

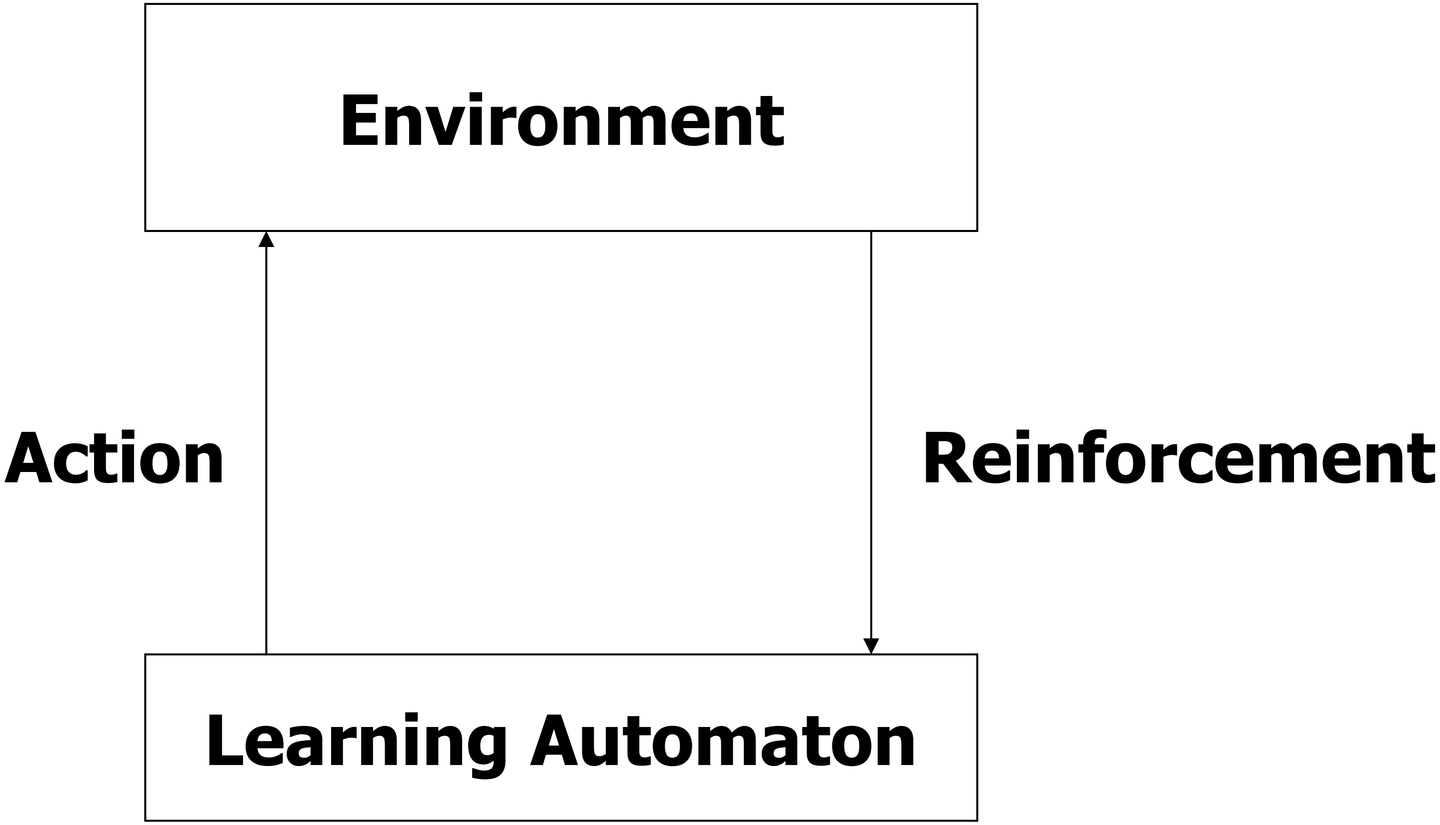
- Learning automaton as a policy iterator
- Overview of Learning Schemes
- Convergence issues

Automata Games

- Definition
- Analytical Results
- Dynamics
- ESRL + Examples

Learning automata

Single Stage, Single Agent



Learning automata

Single Stage, Single Agent

Assume binary feedback, and L actions

When feedback signal is positive,

$$p_i(k+1) = p_i(k) + a\left[1 - p_i(k)\right] \text{ if i}^{\text{th}} \text{ action is taken at time k}$$
$$p_j(k+1) = \left(1-a\right)p_j(k), \text{ for all } j \neq i$$

with a in]0,1[

When feedback signal is negative,

$$p_i(k+1) = \left(1-b\right)p_i(k), \text{ if i}^{\text{th}} \text{ action is taken at time k}$$
$$p_j(k+1) = b/\left(l-1\right) + \left(1-b\right)p_j(k), \text{ for all } j \neq i$$

with b in]0,1[

Reward-penalty, $L_{R\text{-}P}$

Reward-ε penalty, $L_{R\text{-}\varepsilon P}$ $b << a$

Learning automata, cont.

When updates only happen at positive feedback, (or b = 0)

$$p_i(k+1) = p_i(k) + a\left[1 - p_i(k)\right] \text{ if } i^{th} \text{ action is taken at time k}$$
$$p_j(k+1) = (1-a)p_j(k), \text{ for all } j \neq i$$

Reward-in-action, $L_{R\text{-}I}$

Some terminology:

Binary feedback : P-model

Discrete valued feedback: Q-model

Continuous valued feedback : S-model

Finite action Learning Automata : FALA

Continuous action Learning Automata : CALA

General S-model

Reward penalty, $L_{R\text{-}P}$

$$p_i(k+1) = p_i(k) + a \cdot r(k)(1 - p_i(k)) \text{ - b} \cdot (1 - r(k))p_i(k), \text{ with } i \text{ the action taken}$$

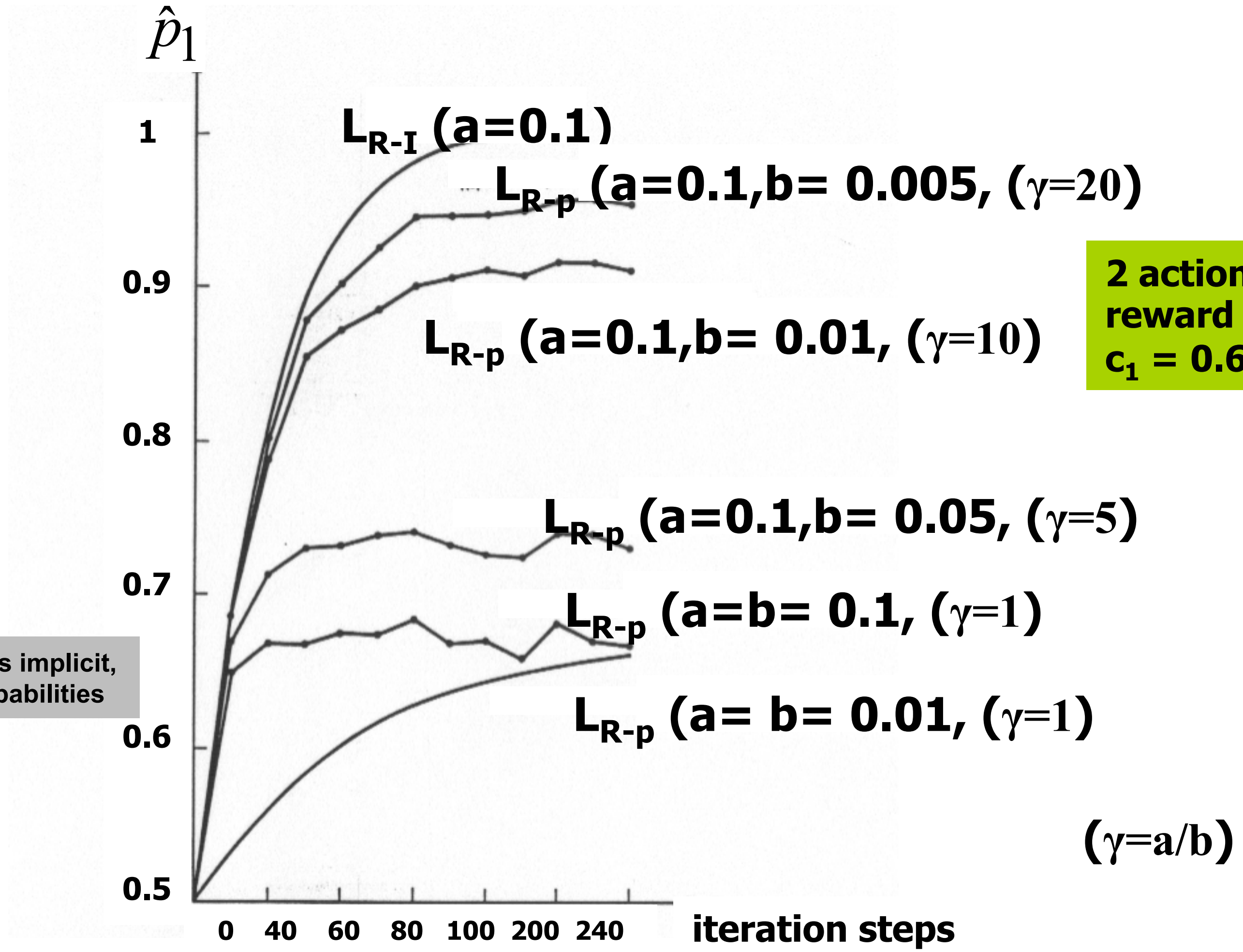
$$p_j(k+1) = p_j(k) - a \cdot r(k)p_j(k) + b \cdot (1 - r(k))\left[(l-1)^{-1} - p_j(k)\right], \text{ for all } j \neq i$$

with r(k) real valued reward signal

If b << a : Reward- ε penalty, $L_{R\text{-}\varepsilon P}$

If b = 0 : Reward-in-action, $L_{R\text{-}I}$

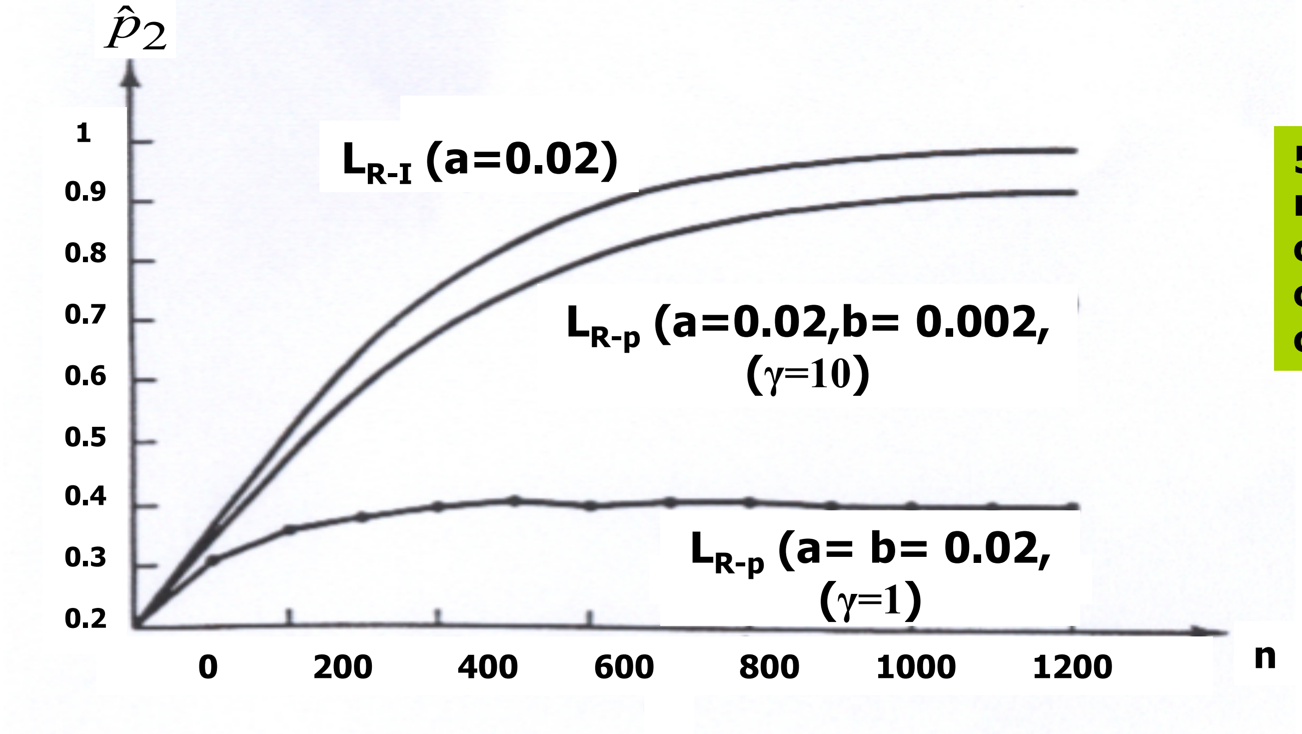
Learning automata, a simulation



2 actions
reward probabilities :
c_1 = 0.6, c_2 = 0.2

Action selection for LA is implicit, based on the action probabilities

Learning automata, a simulation



5 actions
reward probabilities:
c_1 = 0.35, c_2 = 0.8,
c_3 = 0.5, c_4 = 0.6,
c_5 = 0.15

Convergence properties of LA single state, single automaton

$L_{R\text{-}I}$ and $L_{R\text{-}\varepsilon P}$ are ε-optimal in stationary environments:

$$\liminf_{k\to\infty} p_l(k) > 1-\varepsilon', \quad w.p.1.$$

We can make the probability of the best action converge arbitrarily close to 1

$$\lim_{k\to\infty} W(k) > d_l - \varepsilon', \quad w.p.1.$$

We can let the average reward converge arbitrarily close to the highest expected reward

W(K) is the average accumulated reward
Dl the expected reward of the best action

LR-P is not ε-optimal, but Expedient:

$$\lim_{k\to\infty} W(k) > W(0)$$

Performs strictly better than a pure chance automaton

Learning Automata

Basic Definition

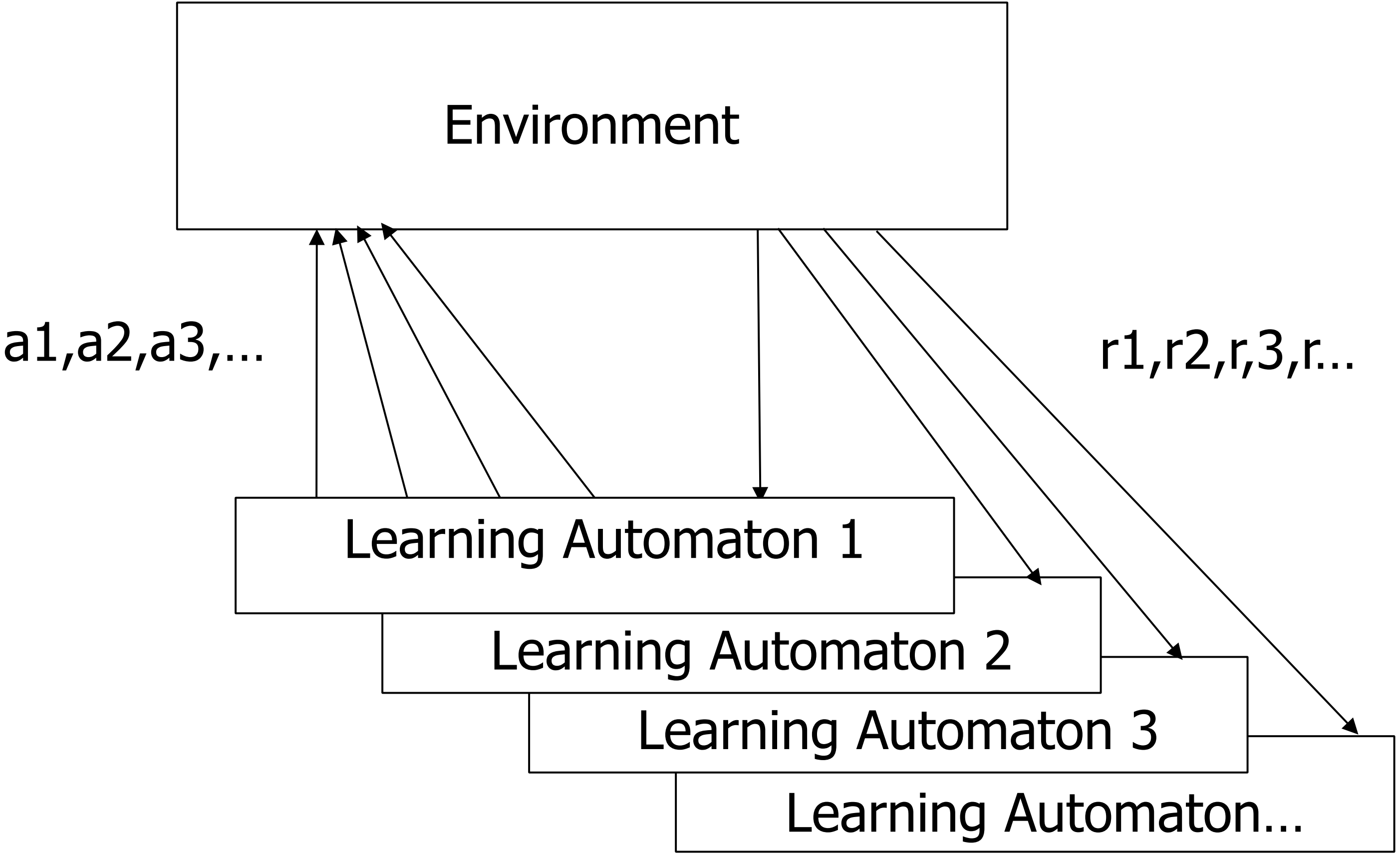
- Learning automaton as a policy iterator
- Overview of Learning Schemes
- Convergence issues

Automata Games

- Definition
- Analytical Results
- Dynamics
- ESRL + Examples

Automata Games

Single Stage, Multi-Automata



Environment

a1,a2,a3,…

r1,r2,r,3,r…

Learning Automaton 1

Learning Automaton 2

Learning Automaton 3

Learning Automaton…

Automata Games

(Narendra and Wheeler, 1989)

Players in an n-person non-zero sum game who use independently a reward-inaction update scheme with an arbitrarily small step size will always converge to a pure equilibrium point.

If the game has a pure NE, the equilibrium point will be one of the pure NE.
Convergence to Pareto Optimal (Nash) Equilibrium not guaranteed.

=> Coordinated exploration will be necessary

Dynamics of Learning Automata

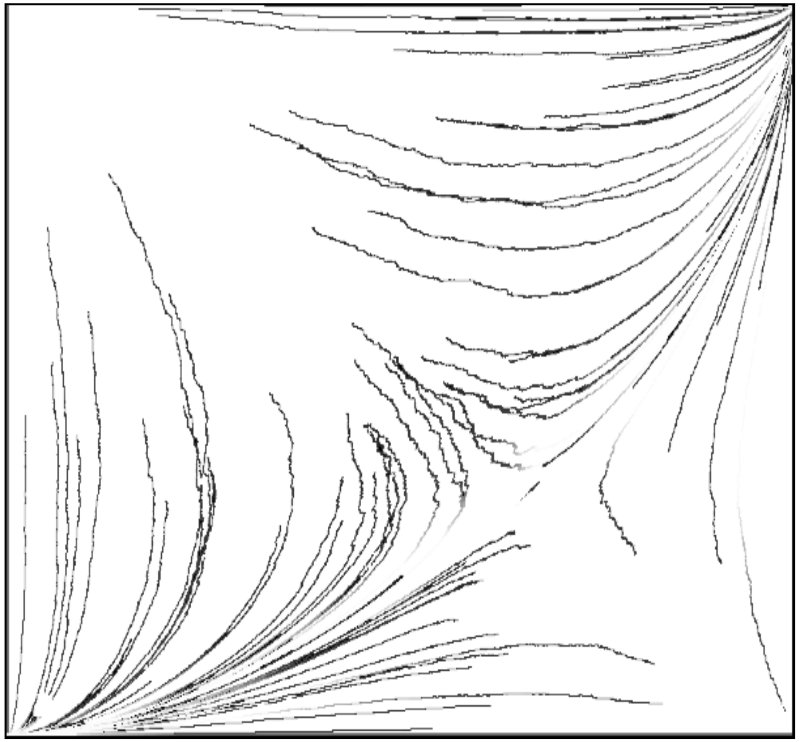
Category 2: Battle of the sexes

Paths induced by a linear reward -inaction LA.

Starting points are chosen randomly

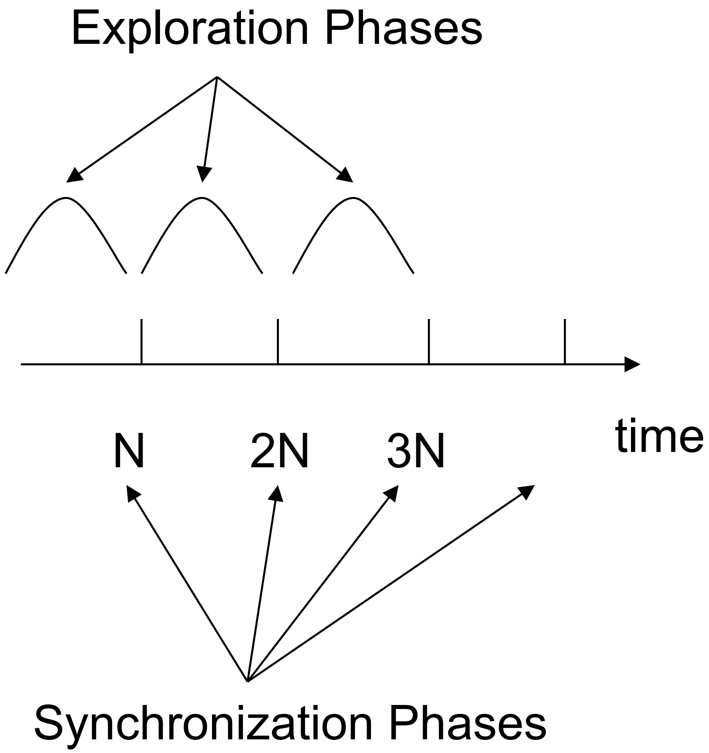
x-axis = prob. of the first player to play Bach

y-axis = prob. of the second player to play Bach



(Tuyls ’04)

Exploring selfish Reinforcement Learners ESRL



Basic idea: 2 phases

- Exploration: Be Selfish
  - Independent Learning
  - Convergence to different NE and Pareto optimal non-NE
- Synchronization: Be Social
  - Exclusion phase: shrink the action space by excluding an action

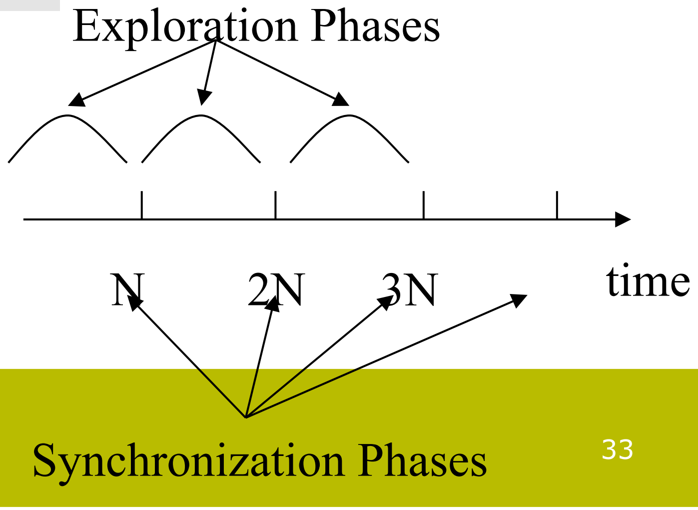
(Verbeeck ’04)

ESRL and common interest games

The Penalty Game

		Player B		
		b_1	b_2	b_3
Player A	a_1	**10,10**	**0,0**	**k,k**
	a_2	**0,0**	**2,2**	**0,0**
	a_3	**k,k**	**0,0**	**10,10**

Witk k < 0



Exploration:

- use L_RI -> the agents converge to a pure (Nash) joint action

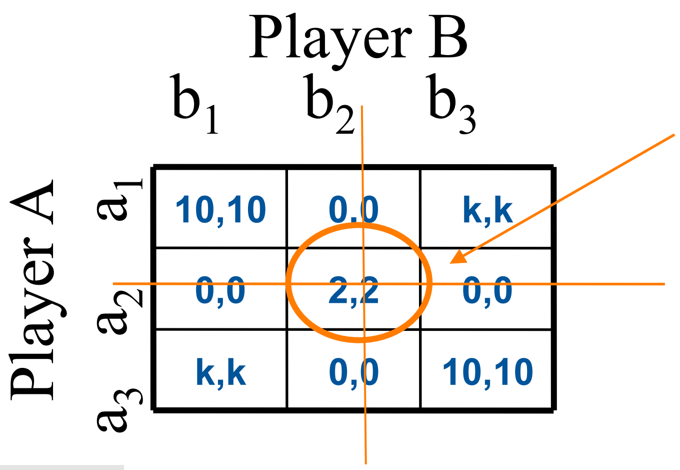
Synchronization:

- update average payoff for action *a* converged to, optimistically
- exclude action *a*, and explore again if empty *action set* -> RESET

If “done”: select BEST

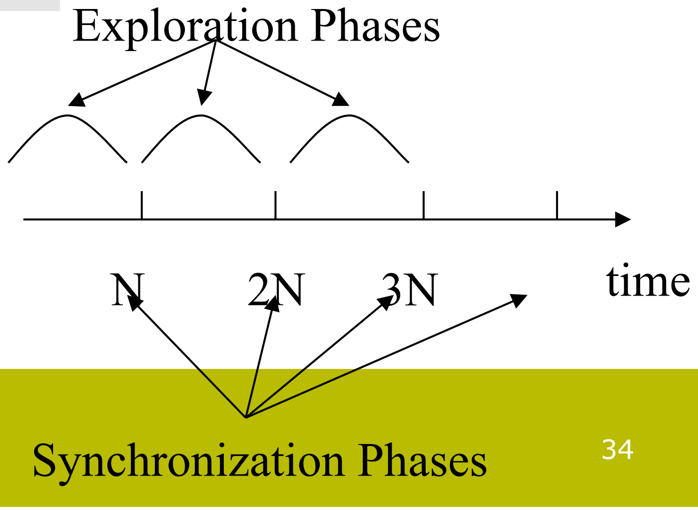
ESRL and common interest games

The Penalty Game



	Player B b_1	Player B b_2	Player B b_3
Player A a_1	10,10	0,0	k,k
Player A a_2	0,0	2,2	0,0
Player A a_3	k,k	0,0	10,10

Witk k < 0



Exploration:

- use L_RI -> the agents converge to a pure (Nash) joint action

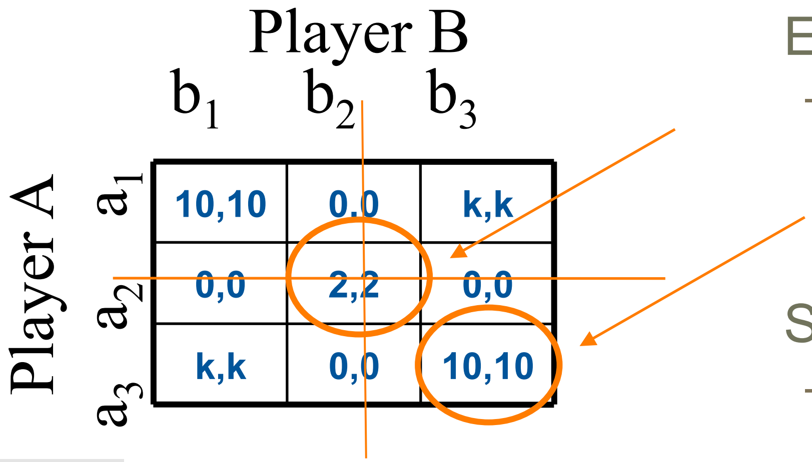
Synchronization:

- update average payoff for action *a* converged to, optimistically
- exclude action *a*, and explore again if empty *action set* -> RESET

If “done”: select BEST

ESRL and common interest games

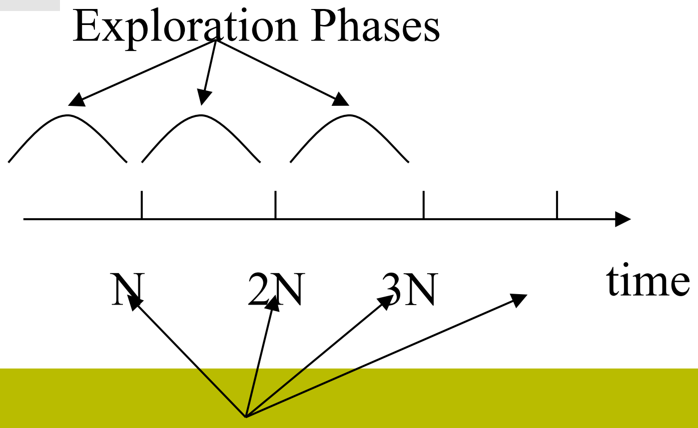
The Penalty Game



Player B

Player A	b_1	b_2	b_3
a_1	10,10	0,0	k,k
a_2	0,0	2,2	0,0
a_3	k,k	0,0	10,10

Witk k < 0



Synchronization Phases

Exploration:

- use L_RI -> the agents converge to a pure (Nash) joint action

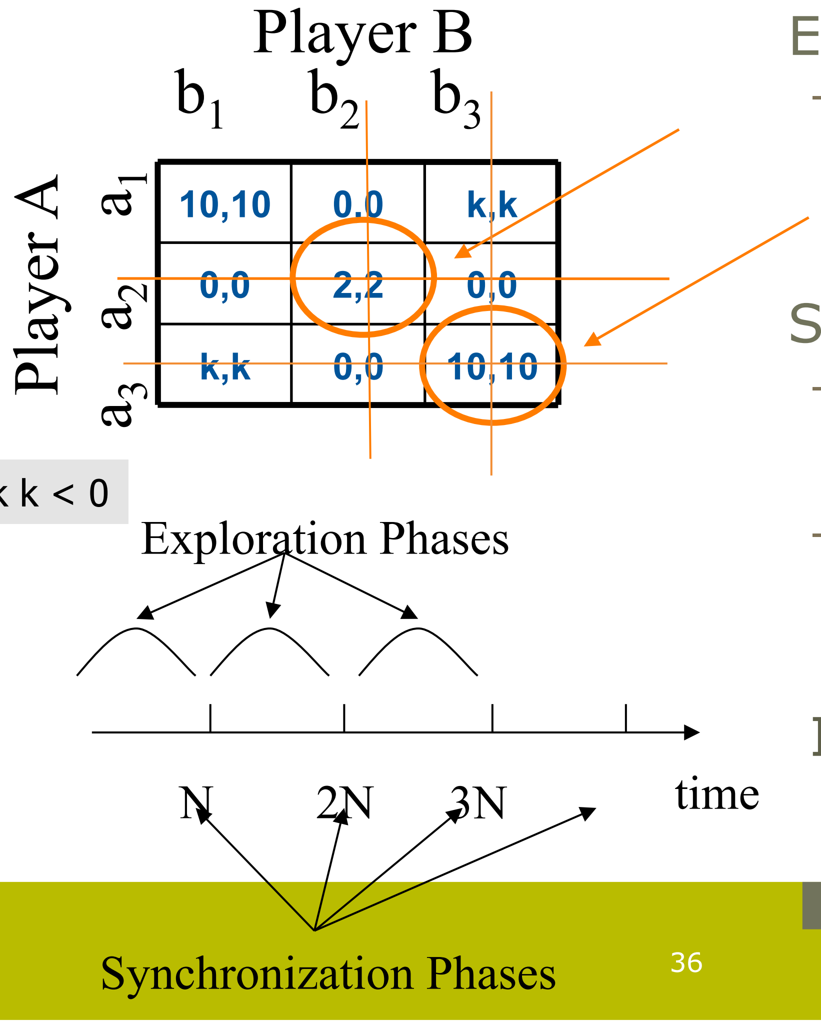
Synchronization:

- update average payoff for action *a* converged to, optimistically
- exclude action *a*, and explore again if empty *action set* -> RESET

If “done”: select BEST

ESRL and common interest games

The Penalty Game



	Player B		
Player A	b_1	b_2	b_3
a_1	10,10	0,0	k,k
a_2	0,0	2,2	0,0
a_3	k,k	0,0	10,10

Witk k < 0

Exploration:

- use L_RI -> the agents converge to a pure (Nash) joint action

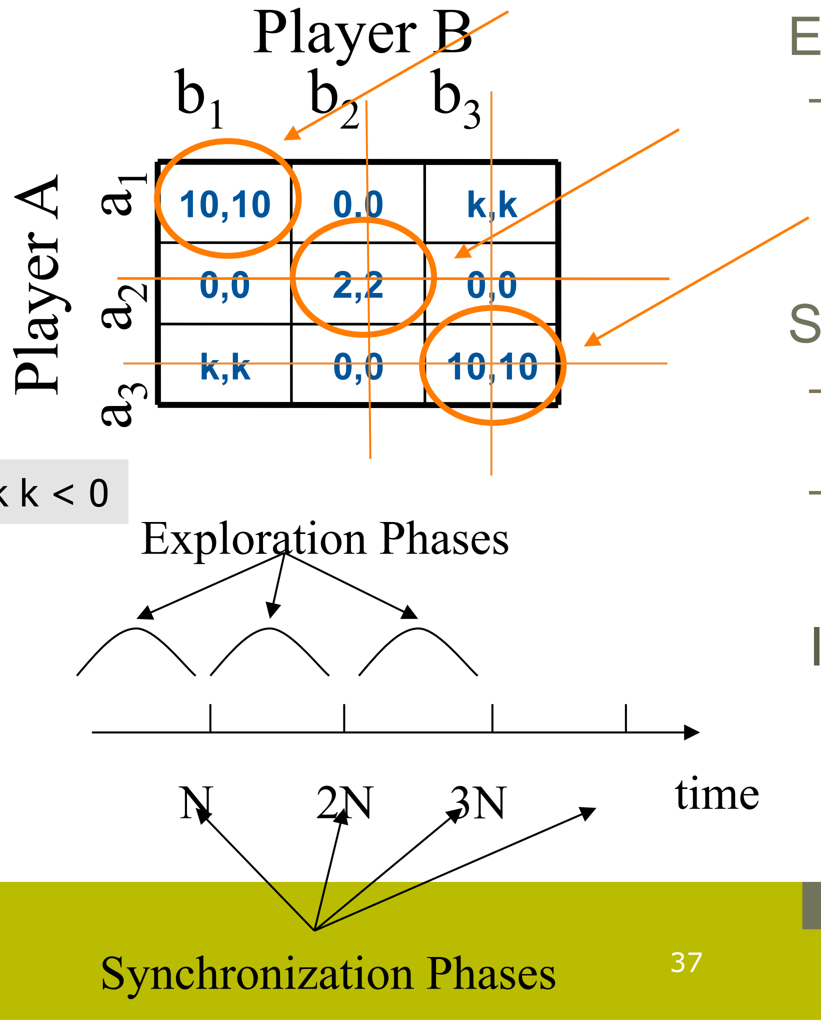
Synchronization:

- update average payoff for action *a* converged to, optimistically
- exclude action *a*, and explore again
  if empty *action set* -> RESET

If "done": select BEST

ESRL and common interest games

The Penalty Game



		Player B		
		b_1	b_2	b_3
Player A	a_1	10,10	0,0	k,k
	a_2	0,0	2,2	0,0
	a_3	k,k	0,0	10,10

Witk k < 0

Exploration:

- use L_RI -> the agents converge to a pure (Nash) joint action

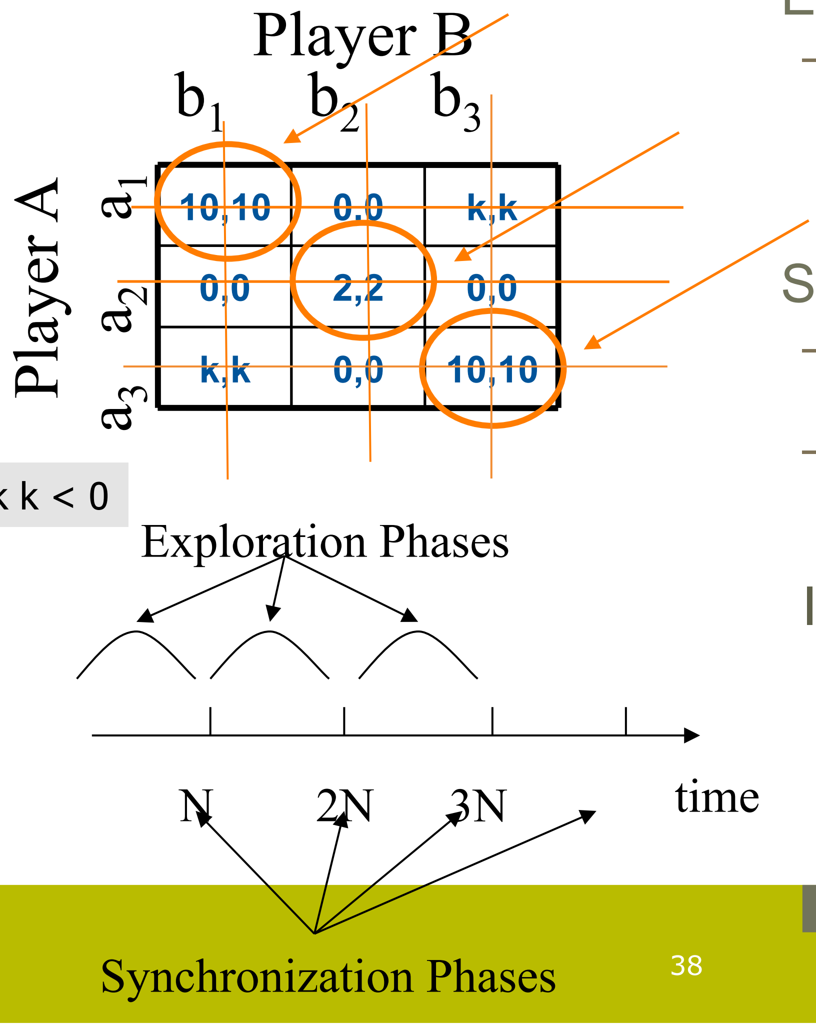
Synchronization:

- update average payoff for action *a* converged to, optimistically
- exclude action *a*, and explore again if empty *action set* -> RESET

If “done”: select BEST

ESRL and common interest games

The Penalty Game



Player B

Player A

	b_1	b_2	b_3
a_1	10,10	0,0	k,k
a_2	0,0	2,2	0,0
a_3	k,k	0,0	10,10

Witk k < 0

Exploration:

- use L_RI -> the agents converge to a pure (Nash) joint action

Synchronization:

- update average payoff for action *a* converged to, optimistically
- exclude action *a*, and explore again if empty *action set* -> RESET

If “done”: select BEST

ESRL and common interest games

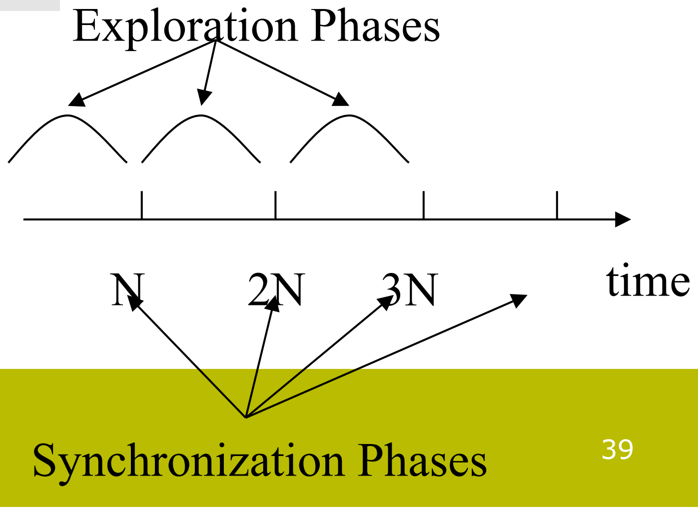
The Penalty Game

Player B

Player A

	b_1	b_2	b_3
a_1	**10,10**	**0,0**	**k,k**
a_2	**0,0**	**2,2**	**0,0**
a_3	**k,k**	**0,0**	**10,10**

Witk k < 0



Exploration:

- use L_RI -> the agents converge to a pure (Nash) joint action

Synchronization:

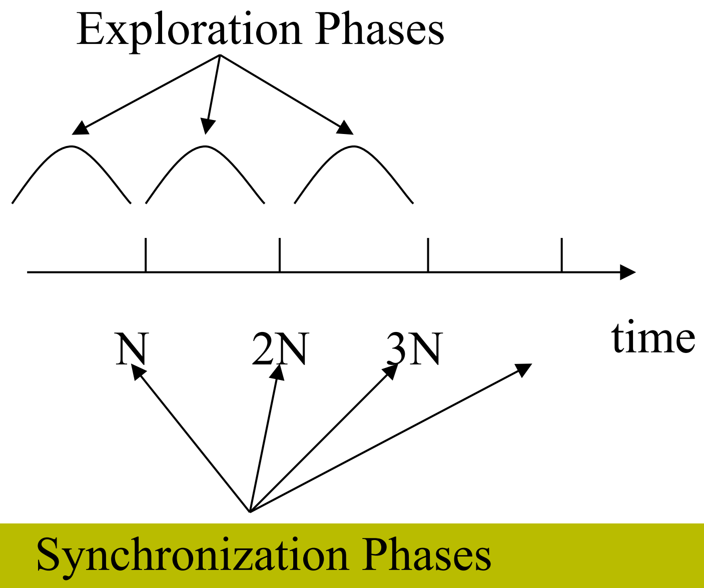
- update average payoff for action *a* converged to, optimistically
- exclude action *a*, and explore again
  if empty *action set* -> RESET

If "done": select BEST

Note : in more than 2 agent games, at least 2 agents have to exclude an action in order to escape from an NE

ESRL and conflicting interest games

	B	**S**
B	**2,1**	**0,0**
S	**0,0**	**1,2**



Exploration:

- use L_RI -> the agents converge to a (Nash) pure joint action

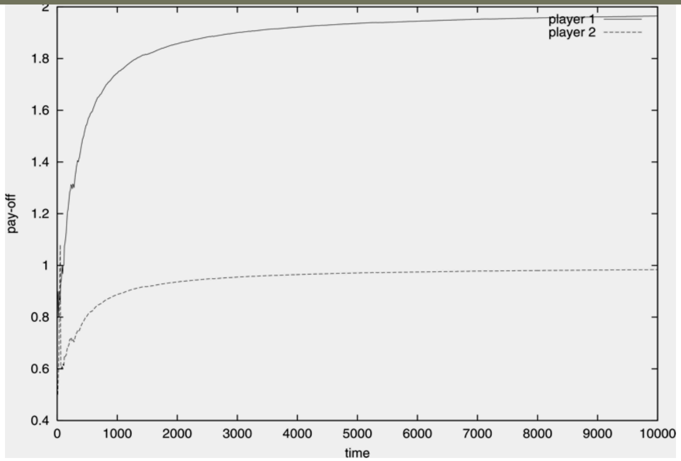
Synchronization:

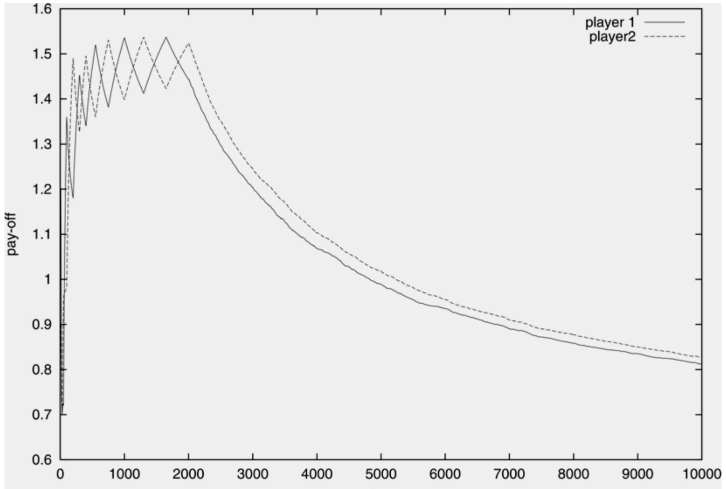
- send and receive average payoff for joint action converged to (not the actions information)
- if best agent : excludes private action
- else RESET

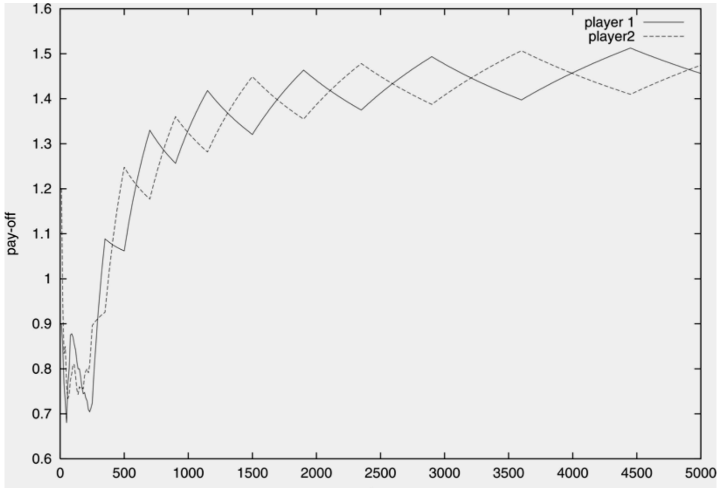
Conflicting Interest games: periodical policies

Player 2

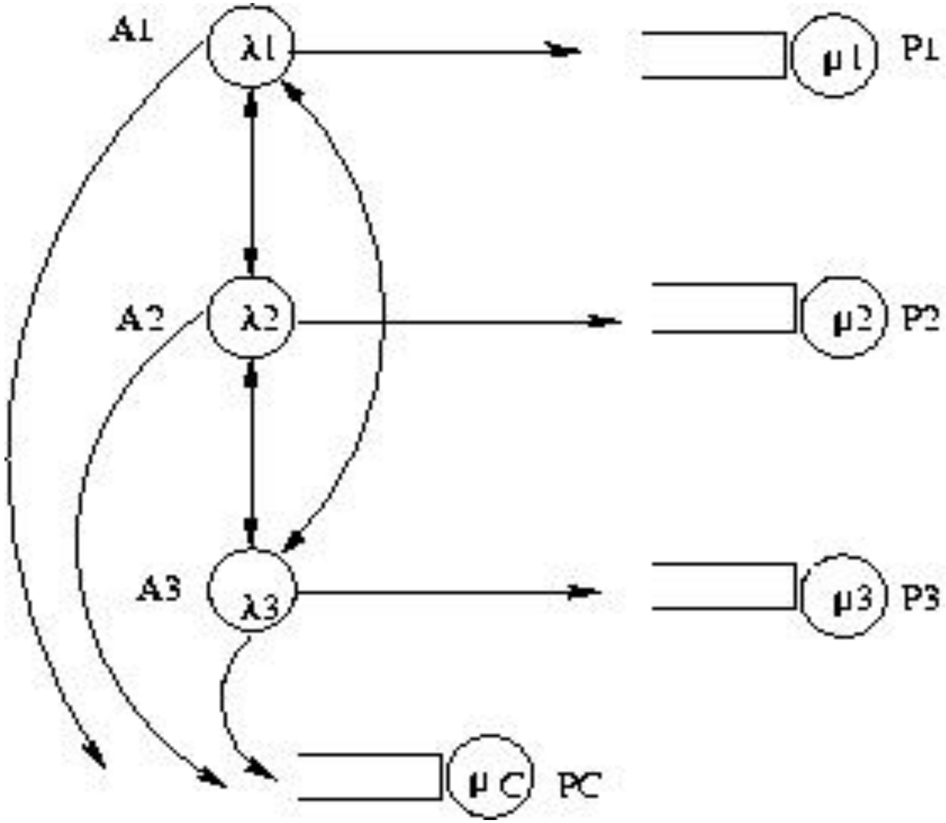
		B	S
Player 1	B	2,1	0,0
	S	0,0	1,2





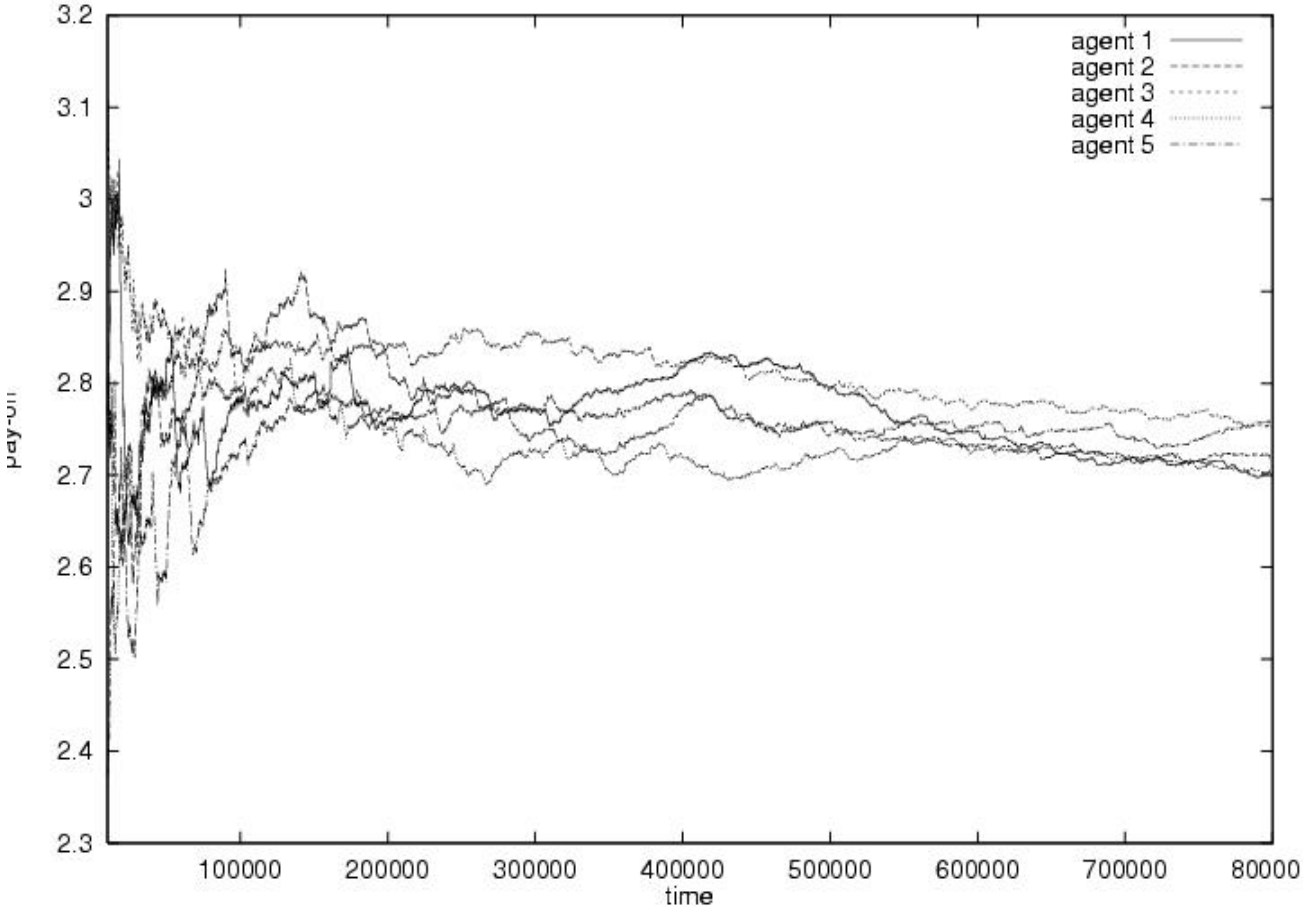


ESRL & Job Scheduling

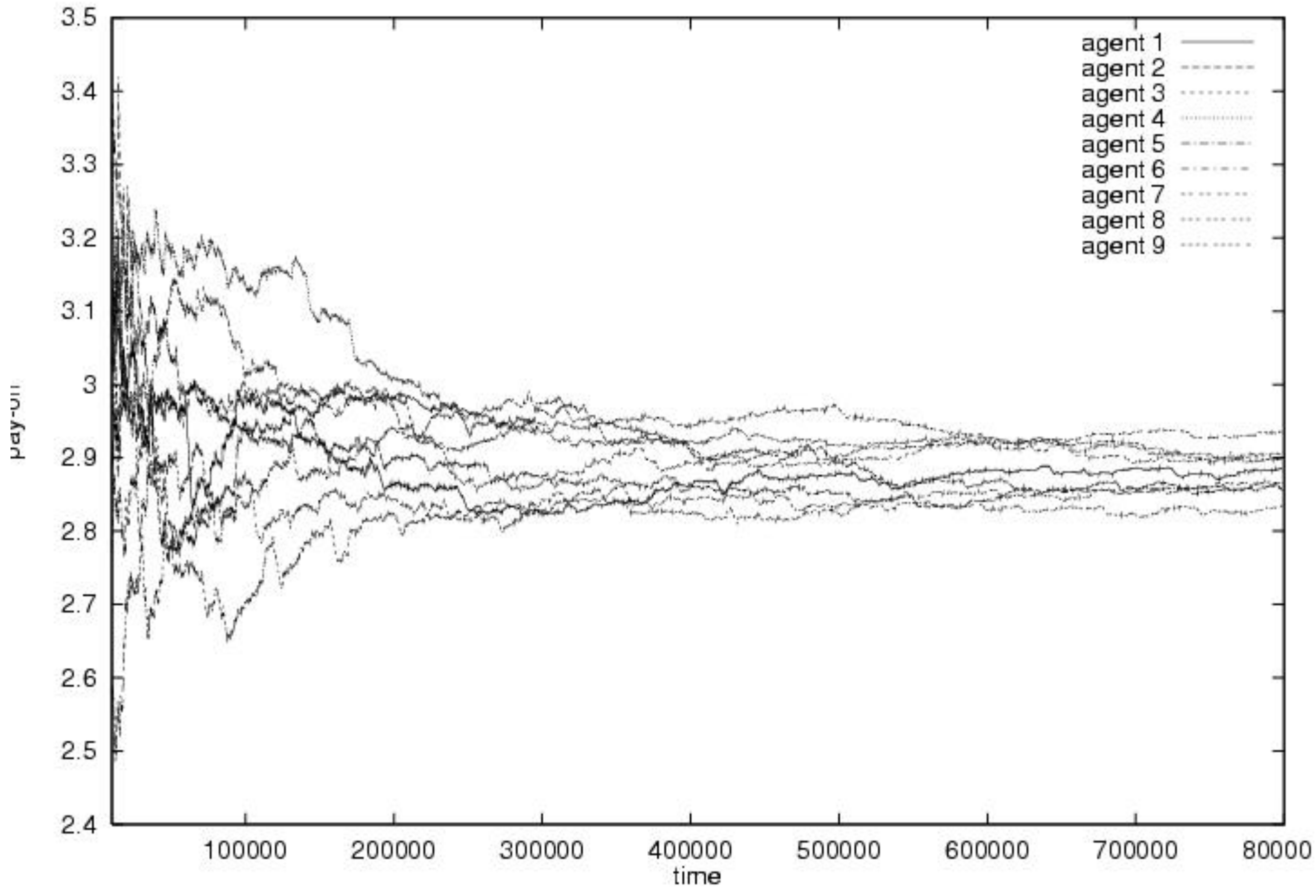


$$\mu 1 = \mu 2 = \mu 3 > \mu C$$

ESRL & Job Scheduling



ESRL & Job Scheduling



To be continued

More next week on graphical games

And interconnected automata
for solving multi-stage problems

References

Nowé, A., Vrancx, P., & De Hauwere, Y. - M. (2012). Game Theory and Multi-agent Reinforcement Learning. In M. Wiering & M. van Otterlo (Eds.), Reinforcement Learning: State of the Art (p. 441-470).

Claus, C., and Boutilier, C. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 746–752.

S. Kapetanakis, D. Kudenko (2004). "Reinforcement Learning of Coordination in Heterogeneous Cooperative Multi-Agent Systems", *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04)*.

S. Kapetanakis, D. Kudenko, M. Strens (2004). "Learning of Coordination in Cooperative Multi-Agent Systems using Commitment Sequences", *Artificial Intelligence and the Simulation of Behavior 1(5)*.

Verbeeck K., Nowé A., Parent J. and Tuyls K., Exploring *Selfish Reinforcement Learning in Stochastic Non-Zero Sum Games,* In The International Journal on Autonomous Agents and Multi-agent Systems., vol.14 (3):239–269, 2007.

Verbeeck K. , Nowé A., Peeters M., Tuyls K., *Multi-Agent Reinforcement Learning in Stochastic Single and Multi-Stage Games*, Adaptive Agents and Multi-Agent Systems II: Editors: D. Kudenko, D. Kazakov, E. Alonso, Lecture Notes in Computer Science, Vol 3394, pp 275-294, 2005.