# Zero-Sum Stochastic Games
# An algorithmic review

Emmanuel Hyon LIP6/Paris Nanterre
with N Yemele and L Perrotin

Rosario November 2017
Final Meeting Dygame
Dygame Project Amstic

# Outline

# Static game definition

A *static game* under a strategic form with complete information is the 3-uple $(\mathcal{N}, \{A_i\}_i, R_i)$ where

- $\mathcal{N}$ is the (finite) set of players (size $N$).
- $A^i$ is the set of action $a^i$ of player $i$ (size $m^i$).
- $R^i(a)$ is the reward of player $i$,
  with $a = (a^1, \dots a^N)$ the set of the actions played by the agents

A *strategy* is said :

- *pure strategy* : when the selection of the action is deterministic.
- *mixed strategy* : when each of the action receive a probability to be chosen :
  in this case $\pi^i(a^i_j)$ is the probability of player $i$ to play $a^i_j$.

# Static game definition II

The *Utility* of agent $i$ is

$$r^i(\pi^i, \pi^{-i}) = \sum_{a^i \in A^i} \sum_{a^{-i} \in A^{-i}} R(a^i, a^{-i}) \pi^i(a^i) \pi^{-i}(a^{-i}). \qquad (1)$$

**Définition (Pure Nash Equilibrium)**

*A set of pure strategies $a^*$ is a Nash Equilibrium if, for all $i$,*

$$R(a^{i*}, a^{-i*}) \geq R(a^i, a^{-i*}) \quad \forall\ a^i \in A^i$$

**Définition (Mixed Nash Equilibrium)**

*A set $\pi^*$ of mixed strategies is a Nash Equilibrium if, for all $i$,*

$$r(\pi^{i*}, \pi^{-i*}) \geq r(\pi^i, \pi^{-i*}) \quad \forall\ \pi^i$$

## Zero-Sum static games

*Zero-Sum game* : the sum of the utilities of all players is null.

*Two players Zero-Sum game* : the reward of a player 1 is equal to the loss of player 2 *i.e.* $\forall a^1, a^2 \quad R^1(a^1, a^2) = -R^2(a^1, a^2)$.

Letting, $r(\pi^1, \pi^2) = r^1(\pi^1, \pi^2)$. In a two players ZS game if $({\pi^1}^*, {\pi^2}^*)$ form a Nash Equilibrium they satisfies

$$r(\pi^1, {\pi^2}^*) \leq r({\pi^1}^*, {\pi^2}^*) \leq r({\pi^1}^*, \pi^2) \; \forall \pi^1 \pi^2 .$$

and are called *Optimal strategies*.

**Théorème (Minimax (Von Neuman))**

*A 2 player ZS Game has a value V if and only if*

$$\max_{\pi^1} \min_{\pi^2} r(\pi^1, \pi^2) = \min_{\pi^2} \max_{\pi^1} r(\pi^1, \pi^2) = V$$

# Static Games and linear Programming

[Filar and Vrieze 96], when solving Minimax equation one can restrict to extreme points :

$$\max_{\pi^1} \min_{\pi^2} \sum_i \sum_j \pi^1(i)\pi^2(j)R(a^1_i, a^2_j) = \max_{\pi^1} \min_{j} \sum_i \pi^1(i)R(a^1_i, a^2_j)$$

Player 1 should then solve

$$\max \min_j \sum_i \pi^1(i)R(a^1_i, a^2_j)$$

s.c.

$$\sum_i \pi^1(i) = 1$$

$$\pi^1(i) \geq 0 \ \forall i \ .$$

$$\max \ v$$

$$\text{s.c. } v \leq \sum_i \pi^1(i)R(a^1_i, a^2_j) \quad \forall \ j$$

$$\sum_i \pi^1(i) = 1$$

$$\pi^1(i) \geq 0 \ \forall i \ .$$

# Stochastic Games description

We assume

- A dynamic game states of which changes over time
- A game different in each state
- Simultaneous actions of players
- A function describes the dynamic evolution of the system w.r.t the simultaneous plays and the state
- When the evolution function is random it is a *stochastic game*.

**Définition (Information Models)**

*Perfect Information The players knows the set of actions, states and rewards until step* $t-1$.

*Closed Loop The player knows the the current state of the game*

## Stochastic Games definition

A stochastic game is a 5-uple $(\mathcal{N}, \mathcal{S}, \mathbf{A}, R, P)$ with :

- $\mathcal{N}$ is the (finite) set of player (size $N$),
- $\mathcal{S}$ is the state space (size $S$),
- $\mathbf{A} = \{A_i\}_{i \in \{1,\dots,N\}}$ is the set of all actions where $A_i$ is the set of actions $a_i$ of player $i$ (size $m^i$),
- $R_i$ is the instantaneous reward of player $i$.
  $R_i(s, a^1, \dots, a^N)$ depends on state and actions of players
- $P$ the transition probability $p(s'|s, a)$ to switch in state $s'$ from $s$ when $a = (a^1, \dots, a^N)$ is played.

Small Taxonomy :
*Stochastic Game* : transition function depends on the history
*Markov Game* : transition function depends on the state
*Competitive Game* : 2 player Zero Sum Markov Game

# Perfect Nash Equilibrium

*Strategy* : the strategy $\pi^i$ of player $i$ is the vector $|S| \times m^i$
$\pi^i = (\pi^i_1, \dots, \pi^i_S)$ with $\pi^i_1$ the mixed strategy on action in state 1.

*Expected utility* $r^i_k(s, \pi)$ is the expected instantaneous reward in $s$ at step $k$ w.r.t $\pi = (\pi^1, \dots, \pi^N)$.

The *Utility* of player $i$ in state $s$ is $v_i(s, \pi)$ ($\gamma$ the discount factor) :

$$v_i(s, \pi) = \mathbb{E}_s \sum_{t=0}^{\infty} {\gamma^i}^t (r^i_k(s, \pi))^t.$$

**Définition (Nash Equilibrium in stochastic game)**

*A set of strategies* $\pi^* = ({\pi^1}^*, \dots, {\pi^N}^*)$ *is a N.E. if,* $\forall s \in S$ *and* $\forall i$ *:*

$$v_i(s, \pi^*) \geq v_i(s, {\pi^1}^*, \dots, {\pi^{i-1}}^*, \ \pi^i \ , {\pi^{i+1}}^*, \dots, {\pi^N}^*) \ \forall \pi^i$$

Interested by *Perfect Nash Equilibrium* = N.E. of any sub-games

# Competitive Games

A competitive game is a 2 players Markov Game.
It is a discounted game
It is a Zero Sum game

$$r^1(s, a^1, a^2) + r^2(s, a^1, a^2) = 0,\ \forall s \in \mathcal{S},\ a^1 \in A^1(s),\ a^2 \in A^2(s).$$

The strategies studied are the Markov Stationary Policies than for static

We have the equivalent definition of optimal strategies

$$v(\pi^1, \pi_0^2) \leq v(\pi_0^1, \pi_0^2) \leq v(\pi_0^1, \pi^2)\ .$$

# Shapley Equation

A competitive game can be seen as a succession of static games each one defines an *Auxiliary matrix game* depending on the state, the strategy and the value function :

$$R(s,v) = \left[ r(s,a^1,a^2) + \beta \sum_{s' \in \mathcal{S}} p(s'|s,a^1,a^2)v(s') \right]_{a^1=1,a^2=2}^{m^1(s),m^2(s)} \tag{2}$$

It follows the Shapley Equation

$$v(s) = \text{val}[R(s,v)]. \tag{3}$$

From [Shapley53] (2 players), [Find 64] ($N$ players) :

- The fix point equation exists and has an unique solution which is called the value vector.
- If the couple $\pi_0^1, \pi_0^2$ is a pair of optimal strategies then $\pi_0^i$ is the stationary optimal strategy of player $i$.

# Outline

# Initial Shapley Algorithm

*Step 1* Start with any $v_0 : \forall s,\ v_0(s)$ has any value

*Step 2*

**Repeat**

for $s \in S$ do :

-Build auxiliary game $R(s, v_{n-1})$

$\left[r(s, a^1, a^2) + \beta \sum_{s' \in \mathcal{S}} p(s'|s, a^1, a^2) v(s')\right].$

-Compute (with Shapley Snow method) the value

and let $v^n(s) = val[R(s, v_{n-1})]$

end for

**until** $\|v_n(s) - v_{n-1}(s)\| < \epsilon\ \forall s$

*Step 3*

for $s \in \mathcal{S}$ do :

- Let $v(s) = v_n(s)$, Build $R(s, v)$
- Compute $\pi^1(s)$ et $\pi^2(s)$ $\pi(s)$ for game $R(s, v)$

end for

**return** $v(s), \pi^1(s), \pi^2(s)\ \forall s.$

# Shapley Algorithm with linear Programming

*Step 1* Start with any $v_0 : \forall s,\ v_0(s)$ has any value

*Step 2*

**Repeat**

for $s \in S$ do :

-Build auxiliary game $R(s, v_{n-1})$

$\left[r(s, a^1, a^2) + \beta \sum_{s' \in \mathcal{S}} p(s'|s, a^1, a^2) v(s')\right]$.

-Compute with LP the value and let $v^n(s) = val[R(s, v_{n-1})]$

$val[R(s, v_{n-1})] = \max_{\pi^1} \min_{a^2 \in A^2} \sum_{a^1} R(s, a^1, a^2)\pi^1(a_1)$.

end for

**until** $\|v_n(s) - v_{n-1}(s)\| < \epsilon\ \forall s$

*Step 3*

for $s \in \mathcal{S}$ do :

- Let $v(s) = v_n(s)$, - Build $R(s, v)$
- Compute (with LP) $\pi^1(s)$ et $\pi^2(s)$ $\pi(s)$ for game $R(s, v)$

end for

**return** $v(s), \pi^1(s), \pi^2(s)\ \forall s$.

# Hoffman Karp Algorithm

*Step 1* Start with approximation $v_0(s) = 0 \quad \forall s$.

*Step 2* At step $n$

Build matrix $R(s, v_{n-1})$

For all $s$,

Find $\pi_n^2(s)$ an optimal strategy of $R(s, v_{n-1})$ for player 2

*Step 3*

For all $s$ solve the MDP

$v_n(s) = \max_{\pi^1} v_\beta(s, \pi^1, \pi_n^2(s))$

*Step 4*

if $\|v_n - v_{n-1}\| > \epsilon$

Then $n = n + 1$ and go to step 2

else stop and return $v = v_n$, $\pi^2 = \pi_n^2$ and $\pi^1$.

# Pollacheck-Avi Itzak Algorithm

*Step 1* Start with arbitrary approximation of $v_0$ :
$\forall s,\ v_0(s)$ has any value.
*Step 2* At step $n$, the value $v_{n-1}$ is known.
For $s \in \mathcal{S}$ do
Build matrix $R(s, v_{n-1})$
Compute the two optimal strategies of game $[R(s, v_{n-1})]$
let $\pi_n^1$ and $\pi_n^2$ be these two strategies
*Step 3*
Compute the value of the game
$v_n = [I - \beta P(\pi_n^1, \pi_n^2)]^{-1} r(\pi_n^1, \pi_n^2)$.
*Step 4*
If $\pi_n^1 = \pi_{n-1}^1$ and $\pi_n^1 = \pi_{n-1}^2$) then stop
else go to step 2

# Remind on Modified Policy Iteration

In Markov Decision Process Framework, *Modified Policy Iteration* is a variant of Policy Iteration that avoid to solve a linear system.

*Step 1* Start with any $v_0$

*Step 2* At step $n$

For all $s$,

Find the optimal deterministic Markov policy

$\pi_n$ is an optimal strategy of game $\tilde{R}(s, v_{n-1})$

*Step 3* (in the classical PI algorithm)

Compute the value of the game

$v_n = [I - \beta P(\pi_n)]^{-1} r(\pi)$.

*Step 3* (in the Modified Policy Iteration)

Approximate the value of the game

$u_0 = v_{n-1}$ Repeat $u_k = \tilde{R}(s, u_{k-1})$

until $k = m$ $v_n = u_m$ *Step 4*

If $\pi_n = \pi_{n-1}$ then stop

else go to step 2

## van der Wal Algorithm (78)

*Step 1* Start with $v_0$ such that $R(s, v_0) \leq v_0(s) \quad \forall s.$

*Step 2* At step $n$

Build matrix $R(s, v_{n-1})$

For all $s$,

Find $\pi_n^2(s)$ an optimal strategy of game $R(s, v_{n-1})$

*Step 3*

For all $s$ approximate the MDP solution

Repeat $m$ times

$\tilde{v} = v_{n-1}$

$\tilde{v}_{n+1}(s) = \max_{\pi^1} \tilde{v}_\beta(s, \pi^1, \pi_n^2(s))$

$v_n = \tilde{v}_m$

*Step 4*

If $\|v_n - v_{n-1}\| > \epsilon$ $n = n + 1$ go to step 2

Else stop and return

# Remind on MDP and Linear Programming

We search $\max_{\pi \in \Pi} v^{\pi}$ satisfying the D.P. equation

$$v(s) = \max_{a} \left( r(s,a) + \beta \sum_{s' \in \mathcal{S}} p(s'|s,a) v(s') \right), \ \forall s \in \mathcal{S} .$$

Since ($L$ is the Bellman Operator), if $v \geq Lv$ then $v \geq v^*$ and then $\sum_s v(s) \geq \sum_s v^*(s)$. We can solve the problem by minimizing the sum insuring the respect of the constraints $v \geq Lv$.
We get the primal [Filar96]

$$\min_{v \in \nu} \sum_{s=1}^{S} \frac{1}{S} v(s) \qquad (P_\beta)$$

with the set of constraints :

$$v(s) \geq r(s,a) + \beta \sum_{s'=1}^{S} p(s'|s,a) v(s'), \quad \forall a \in A(s), \forall s \in \mathcal{S}.$$

# Single Controller Game

We consider a game in which transitions are controlled only by player 1. It has the property than

$$p(s'|s, a^1, a^2) = p(s'|s, a^1), \tag{4}$$

for all $s,\ s' \in \mathcal{S},\ a^1 \in A^1(s), a^2 \in A^2(s)$.

*Fact 1.* In the game $[R(s, v)]$, the coordinate with index $a^1, a^2$ can be expressed by :

$$r(s, a_1, a_2) + \beta \sum_{s' \in S} p(s'|s, a^1) v(s') .$$

*Fact 2.* With the optimal strategies Equation, we have

$$v(\pi^1(s), \pi_0^2(s)) \leq v(\pi_0^1(s), \pi_0^2(s))$$

for any $\pi^1(s)$ and namely for all pure strategies (*i.e. actions*).

# Single Controller Game (Primal)

Fact 1 and Fact 2 gives

$$v_\beta \geq \sum_{a^2} \pi_0^2(s, a^2) r(s, a^1, a^2) + \beta \sum_{s' \in S} p(s'|s, a^1) v_\beta(s') \, \forall s, a^1.$$

This leads to the Primal formulation

$$\min \sum_{s'=1}^{S} \frac{1}{S} v(s^{'}) \qquad (P_\beta(1))$$

under constraints :

(a) $v(s) \geq \sum_{a_2=1}^{m_2(s)} r(s, a^1, a^2) \pi^2(s, a^2) +$
$\beta \sum_{s'=1}^{S} p(s'|s, a^1) v(s'), \ \forall s \in \mathcal{S}, \ \forall a^1 \in A^1(s),$

(b) $\sum_{a^2 \in A^2(s)} \pi^2(s, a^2) = 1, \ \forall s \in \mathcal{S},$

(c) $\pi^2(s, a^2) \geq 0, \ \forall s \in \mathcal{S}.$

# Single Controller Game (Dual)

$$\max \sum_{s=1}^{S} z(s) \qquad (D_\beta(1))$$

under constraints :

$$\text{d)} \sum_{s=1}^{S} \sum_{a^1 \in A^1(s)} [\delta(s,s^{'}) - \beta p(s^{'}|s,a^1)] x_{s\ a^1} = \frac{1}{S},\ \forall s^{'} \in \mathcal{S}\,,$$

$$\text{e)}\ z(s) \leq \sum_{a^1=1}^{m^1(s)} r(s,a^1,a^2)\mathrm{x}(s,a^1)\,,\ \forall s \in S,\ \forall a^2 \in A^2(s)\,,$$

$$\text{f)}\ x(s,a^1) \geq 0,\ \forall s \in \mathcal{S},\ \forall a^1 \in A^1(s)\,.$$

with $\mathrm{x}(s) = (x(s,1), x(s,2), ..., x(s,m^1(s)))\ \forall s \in \mathcal{S}$.

Theorem 3.2.1 of [Vrieze96] insures that from the solutions of the primal and the dual we obtain the value and the optimal strategies.

# Other Model

There is other models for which linear programming works :

- *Separable reward and transition independent of the state*
- *Switching Controller Game*
  - M1 Transform it in a single controller
  - M2 Solve successive alternates of primal and dual problems

# Extension

For a general model, this does not extend.
Indeed since *fact 1* does not occur then *Fact2* becomes

$$v_\beta \geq \sum_{a^2} \pi_0^2(s,a^2) r(s,a^1,a^2) + \beta \sum_{s' \in S} \sum_{a^2} \pi_0^2(s,a^2) p(s'|s,a^1,a^2) v_\beta(s')$$
$$\forall s, a^1$$

This is not linear but *bilinear*. This is a Non Linear Problem (NLP). So, no method of LP applies.

However, we have two NLP (one for each player) and we can express a single NLP solutions of which are the value of the game and the stationary policies.

Theoretically interesting but hard to solve numerically.

# Reinforcement Learning

Reinforcement learning algorithms to learn equilibrium are base on the *Q learning* (Sutton 1994) Method.

The seminal algorithm is from Litman in 1994. It learns value function with Q learning method and solves some static zero sum games at each iteration.

It has been improved by Nash Q framework