iF 122 5/5/21

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

# Jeux stratégiques / matriciels

- plusieurs joueurs
- jeux concurrents (à la pierre - feuille - ciseaux)
- gain / pertes, équilibres de Nash
- pas nécessairement à somme nulle
- jeux non-coopératifs

Joueurs $\{1, \dots, n\}$

Def Jeu stratégique $G = (S_1, \dots, S_n, p_1, \dots, p_n)$

- $S_i$ = ens. fini de stratégies pour joueur $i$
- $p_i$ : payoff, $p_i : S_1 \times \dots \times S_n \to \mathbb{R}$ du joueur $i$

Les stratégies sont choisies simultanément.

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

On note $S = (S_1, \dots, S_n)$

$1 \leq i \leq n$ $\quad S_{-i} = (S_1, \dots S_{i-1}, S_{i+1}, \dots S_n)$

Les joueurs sont rationnels : ils veulent maximiser leur gain individuel ; ils connaissent les $S_i$, $p_i$.

Ex 1) Prisoner's dilemma

$S_1 = S_2 = \{C, D\}$

C : coopérer (parler) D : défect (se taire)

| 1 \ 2 | C | D |
|---|---|---|
| C | $-2, -2$ | $0, -3$ |
| D | $-3, 0$ | $-1, -1$ |

eq de Nash

$p_1(C, D) = 0$

$p_2(C, D) = -3$

Pareto efficient

2) Battle of sexes (Bach vs. Stravinsky)

| 1 \ 2 | F | B |
|---|---|---|
| F | $2, 1$ | $0, 0$ |
| B | $0, 0$ | $1, 2$ |

1) et 2) : pas somme nulle

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

3) Matching pennies

| | H | T |
|---|---|---|
| H | 1, −1 | −1, 1 |
| T | −1, 1 | 1, −1 |

Jeu à somme nulle

pas d'eq. de Nash

On essaie d'identifier les meilleures stratégies :

- Si s'appelle "meilleure réponse" par rapport à une stratégie $s_{-i}$ des autres si

$$\forall s_i' \in S_i : \quad p_i(s_i, s_{-i}) \geq p_i(s_i', s_{-i})$$

- $s = (s_i)$ s'appelle <u>équilibre de Nash</u> si chaque $s_i$ est "meilleure réponse" à $s_{-i}$.

(personne n'a l'intérêt de changer la stratégie de son côté)

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

- on appelle $s = (s_i)_i$ : Pareto efficient si pour aucune $s' = (s'_i)_i$ on a :

$\forall i \quad p_i(s') \geq p_i(s)$ et

$\exists i \quad p_i(s') > p_i(s)$

## Stratégies dominantes

- $s_i, s_i' \in S_i$

$s_i$ domine strictement $s_i'$ si

$$\forall s_{-i} \in S_{-i} : p_i(s_i, s_{-i}) > p_i(s_i', s_{-i})$$

$s_i$ domine faiblement $s_i'$ si

$$\forall s_{-i} \in S_{-i} : p_i(s_i, s_{-i}) \geq p_i(s_i', s_{-i})$$

et $\exists s_{-i} \in S_{-i}$ :

$$p_i(s_i, s_{-i}) > p_i(s_i', s_{-i})$$

$s_i$ domine $s_i'$ si

$$\forall s_{-i} \in S_{-i} : \quad p_i(s_i, s_{-i}) \geq p_i(s_i', s_{-i})$$

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

- $s_i$ s'appelle strictement / faiblement dominante si pour toute $s_i' \neq s_i$ :
  $s_i$ domine strictement / faiblement $s_i'$

Rq Un joueur rationnel ne va jamais choisir une stratégie qui est strictement dominée.

Prisoners' dilemma :

C ? D

C est strictement dominée par D

→ le choix rationnel est de jouer D.

Rq Supposons que $s = (s_i)$ est tq. chaque $s_i$ est dominante. Alors $s$ est équilibre de Nash.

Car $p_i(s_i, s_{-i}) \geq p_i(s_i', s_{-i})$ , $\forall i$

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

Ex

| 1 \ 2 | L | R |
|---|---|---|
| T | 1,1 | 1,1 |
| B | 1,1 | 0,0 |

T ? B  T domine faiblement B,
L domine faiblement R

Si on enlève les stratégies faiblement dominées, donc B et R, on obtient (T,L) : équilibre de Nash

## Algorithmes pour calculer des équilibres de Nash (s'il y en a)

① Elimination des stratégies dominées strictement (IESDS - iterative elimination of strictly dominated strategies).

$G = (S_1, \dots, S_n, p_1, \dots, p_n)$
→ restrictions $R_i \subseteq S_i$
$\emptyset \neq$

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

$G' = (R_1, \dots, R_n, p_1, \dots, p_n)$ restriction de $G$

$G \to G'$ si $\forall i$ $R_i \subseteq S_i$ est tq :

$\forall s_i \in S_i \setminus R_i$ $\exists s_i' \in R_i$ tq. $s_i$ est strictement dominée par $s_i'$

Ex

| 1 \ 2 | L | M | R |
|---|---|---|---|
| T | 3,0 | 2,1 | 1,0 |
| C | 2,1 | 1,1 | 1,0 |
| B | 0,1 | 0,1 | 0,0 |

G

unique eq. de Nash

B ? T  T domine str. B

R ? M  M ——— R

On élimine B, R :

| | L | M |
|---|---|---|
| T | 3,0 | 2,1 |
| C | 2,1 | 1,1 |

T domine strict. C

| | L | M |
|---|---|---|
| T | 3,0 | 2,1 |

On élimine C : .. On élimine L

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

**iESDS** : on élimine les stratégies strict. dominées, tant que c'est possible.

Si à la fin on obtient exactement une stratégie pour chaque joueur, alors on dit que iESDS résout le jeu G.

Le jeu G' obtenu par iESDS à partir de G s'appelle résultat de iESDS sur G.

**Th (iESDS)**

Soit $G' = (R_1, \dots, R_n, p_1, \dots, p_n)$ un résultat de iESDS sur $G = (S_1, \dots, S_n, p_1, \dots, p_n)$.

1) Si $s$ est eq. de Nash pour $G$, alors $s$ est eq. de Nash pour $G'$.

2) Si $G$ est fini et $s$ est eq. de Nash de $G'$, alors $s$ est eq. de Nash de $G$.

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

3) Si $G$ est fini et résolu par IESDS, alors l'unique stratégie restante est l'unique eq. de Nash pour $G$.

## démo

1) soit $s$ eq. de Nash pour $G$

Que sait-on sur les $s_i$ ?
$s_i$ ne peut pas être strict. dominée par une autre stratégie $s_i'$ (Nash)
Donc $s_i$ ne sera jamais éliminée.

2) Supposons que $G \to G'$ en éliminant une stratégie $s_i'$, parce que $s_i'$ est strict. dominée par $s_i''$
Prenons $s = (s_i)_i$ eq. de Nash de $G'$.

$$p_i(s_i', s_{-i}) \overset{\text{str.d.}}{<} p_i(s_i'', s_{-i}) \overset{\text{Nash de } G'}{\leqslant} p_i(s_i, s_{-i})$$

Donc $s = (s_i)_i$ est eq. de Nash de $G$.

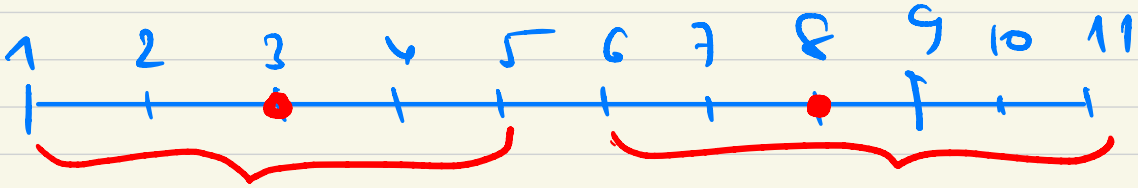
Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

Ex Location game (Hotelling, 1929)

Joueurs : vendeurs qui choisissent une adresse. Les clients vont s'adresser au vendeur qui est le plus proche :

Adresses : $i \in \{1, \dots, n\}$

A chaque adresse se trouve exact. 1 client.




$p(3,8) = (5,6)$

$p_1(3,8) = 5$, $p_2(3,8) = 6$

Si un client est à dist. égale des 2 vendeurs → on partage le gain 1/2

$$p_i(s_i, s_{3-i}) = \begin{cases} \dfrac{s_i + s_{3-i} - 1}{2} & s_i < s_{3-i} \\ n - \dfrac{s_i + s_{3-i} - 1}{2} & s_i > s_{3-i} \\ \dfrac{n}{2} & s_i = s_{3-i} \end{cases}$$

$i \in \{1, 2\}$

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

$$p_i(s_i, s_{3-i}) = \begin{cases} \dfrac{s_i + s_{3-i} - 1}{2} & s_i < s_{3-i} \\ n - \dfrac{s_i + s_{3-i} - 1}{2} & s_i > s_{3-i} \\ \dfrac{n}{2} & s_i = s_{3-i} \end{cases}$$

$i \in \{1, 2\}$

$s_1 = 1$ , $s_2 \in \{1, \dots, n\}$

$p_1(1, s_2)$ $\qquad$ $p_1(2, s_2)$

$$= \begin{cases} s_2/2 & \text{si } s_2 \neq 1 \\ n/2 & \text{sinon} \end{cases} = \begin{cases} (s_2+1)/2 & 2 < s_2 \\ n/2 & 2 = s_2 \\ n-1 & s_2 = 1 \end{cases}$$

$s_1 = 1$ est str. dom. par $s_1' = 2$

$s_1 = n$ $\rightarrow$ ——— par $s_1' = n-1$

$n = 2k+1$ : le jeu est résolu par IESDS en $k$ rounds, et la stratégie qui survit est $s = (k, k)$. $\rightarrow$ unique eq. de Nash du jeu.

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

Q: Est-ce que l'ordre pour éliminer les strat. strict. dominées est important ?

Non : étant donné un jeu fini, l'algorithme iESDS produit le même résultat, quelque soit l'ordre de l'élimination.

Rq. L'hypothèse fini est importante aussi pour les conclusions (2) et (3)

Ex $S_i = \mathbb{N} = \{0, 1, \ldots\}$

$p_i(s) = s_i$

$\forall s_i \quad \exists s_i'$ tq. $s_i$ est str. dominé par $s_i'$

$S_i$ iESDS

- élimine toutes les strat. str. dominées d'un coup $\rightarrow \emptyset$

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

- élimine toutes les stratégies strict-dominées sauf $s_i = 0, \forall i$
→ un unique résultat $s = 0$, n'est pas éq. de Nash

- élimine les st. strict. dominées une par une, alors l'algo ne termine pas

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

$G = (V_0 \dot\cup V_1, E, p : \{0, \dots, d\})$

jeu de parité

$\pi = v_0, v_1, \dots$ gagné par $\underline{P_0}$ si

(*) $\max(\alpha : \exists^\infty n : p(v_n) = \alpha)$

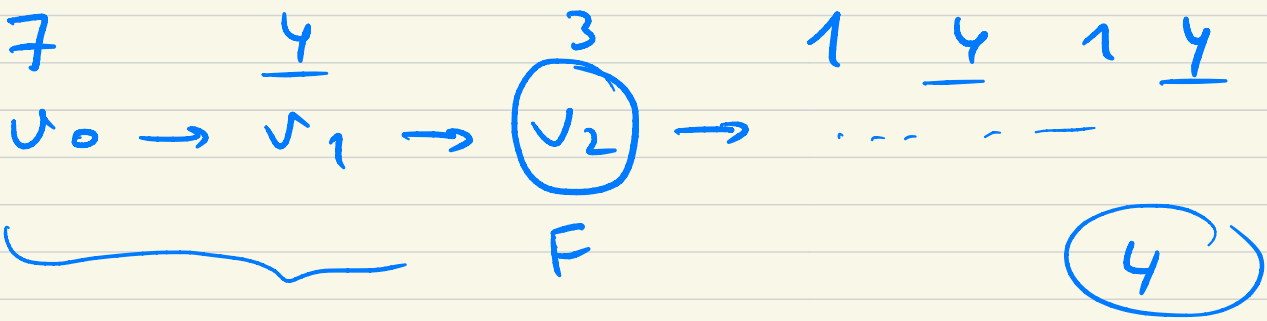
est pair

$G$ + $F \subseteq V$

$\pi = v_0, v_1, \dots$ gagné par $P_0$ si

(*) et $\exists n : v_n \in F$

Parity $\wedge$ Reach(F)




$\pi = v_0, v_1, \dots, v_i, \dots$

$\pi[i] = v_i, v_{i+1}, \dots$

$\pi$ est gagné pour Parity par $P_0$ $\Leftrightarrow$ $\pi[i]$

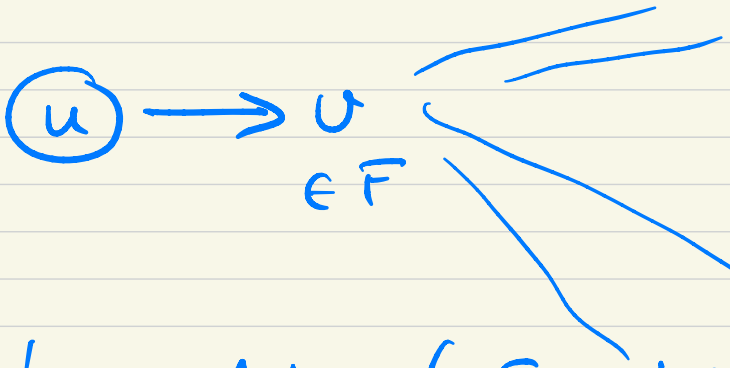
Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

1. Idée → on calcule $W_0, W_1$ pour le jeu parité normal

$F \cap W_0$

= ens. des sommets de F à partir desquels $P_0$ gagne le jeu de parité

$v \in F \cap W_0$




$W_0' = Attr_0(F \cap W_0)$

$\supseteq$ $P_0$ gagne en jouant la stratégie d'atteindre vers $F \cap W_0$, puis la strat. de jeu de parité

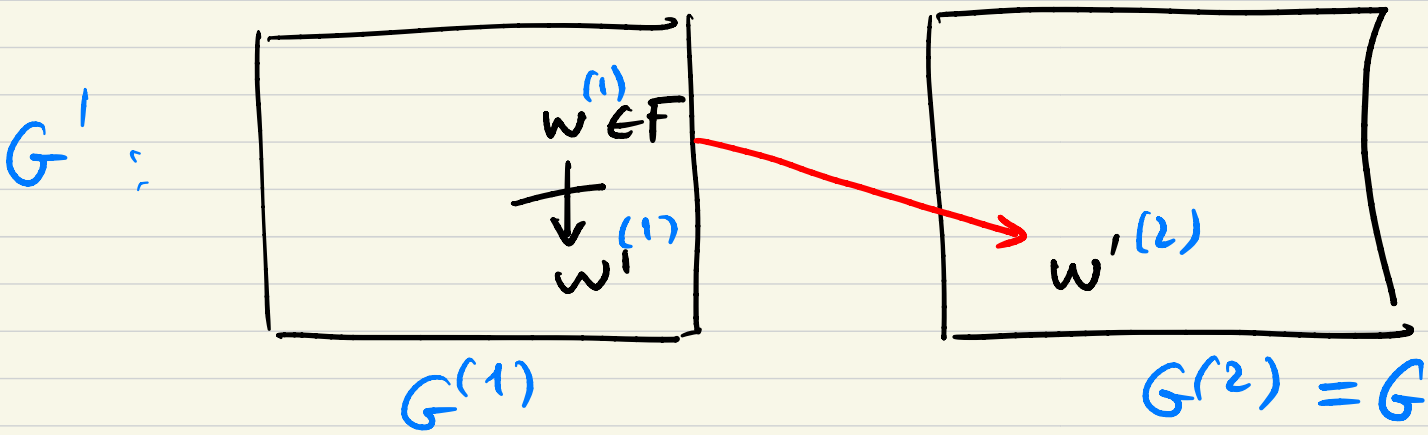
$\subseteq$ ?

! $v \notin Attr_0(F \cap W_0) \Rightarrow P_1$ gagne

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

## 2. Idée.

On construit 2 copies de l'arène de jeu :




$G^{(1)}$ = $G$ \ arêtes sortantes de $F$

$G'$ : jeu de parité

$p'(v^{(1)}) = \cancel{0}\ 1$

$p'(v^{(2)}) = p(v)$

$v^{(1)}$ est gagnant dans $G'$ $\Leftrightarrow$ $v$ ——→ —— dans Parity ∩ Reach(F)

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0