

# Networking modelling

2 mars 2013

Page du cours : <http://www.liafa.univ-paris-diderot.fr/~mairese/MPRI>

## Table des matières

<b>I</b>	<b>Markov Decision Processes</b>	<b>2</b>
<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Horizon fini</b>	<b>3</b>
<b>3</b>	<b>Coûts discounted dans le cadre horizon infini</b>	<b>5</b>
3.1	Notations vectorielles . . . . .	5
3.2	Équation d’optimalité . . . . .	6
3.3	Algorithme par itération de valeur (VIA) . . . . .	7
3.4	Algorithme par itération de stratégie (PIA) . . . . .	8
3.5	Exemple . . . . .	8
3.6	Stratégies structurées . . . . .	9
<b>4</b>	<b>Récompense moyenne</b>	<b>10</b>
4.1	Critère d’optimalité . . . . .	10
4.2	Processus de récompense Markovien (MRP) . . . . .	11
4.3	Équation d’évolution . . . . .	12
4.4	Classification des Processus de décision Markovien (MDP) . . . . .	12
4.5	Équation d’optimalité pour les unichânes . . . . .	12
<b>II</b>	<b>Réseaux de requêtes Markoviennes - Product Forms</b>	<b>13</b>
<b>5</b>	<b>Model de requêtes en temps discret</b>	<b>14</b>
<b>6</b>	<b>Real-life network</b>	<b>15</b>
6.1	Processus Markovien . . . . .	15
6.2	Processus de Poisson . . . . .	15
6.3	Processus de Markov . . . . .	16
6.4	Infinitesimal generator . . . . .	16
6.5	Du modèle au générateur infinitésimal . . . . .	16
6.5.1	Loss network . . . . .	18
6.5.2	Tandem network . . . . .	18
6.5.3	Réseaux de Jackson . . . . .	19
6.6	Réseau de Gordon-Newell . . . . .	22

<b>III</b>	<b>Game theory</b>	<b>22</b>
6.7	$M/M/1/\infty$ : . . . . .	22
<b>7</b>	<b>Finite games (atomic games)</b>	<b>23</b>
<b>8</b>	<b>Non-atomic games</b>	<b>24</b>
8.1	Jeu à populations multiples . . . . .	24
8.1.1	Jeux de congestion . . . . .	24
8.1.2	Jeux à potentiel . . . . .	24
<b>IV</b>	<b>Price of anarchy in multiserver queue</b>	<b>25</b>

## Première partie

# Markov Decision Processes

(3 cours / bruno.gaujal@inria.fr / basé sur le livre *Markov decision processes*)

## 1 Introduction

On se place dans un cadre avec un temps discret  $t \in \mathbb{N}$  et un espace d'état  $S$  (parfois fini). On appelle  $X_0$  l'état au temps  $t = 0$ , puis l'état évolue selon une dynamique markovienne. On a un espace d'action  $\mathcal{A}$  que l'on suppose fréquemment compact et  $\forall a \in \mathcal{A}$ , on a  $P(i, a, j)$  la probabilité d'aller de l'état  $i$  à l'état  $j$  sous l'action de  $a$ .

$$\mathbb{P}(X_{t+1} = j \mid X_t = i, A_t = a) = P(i, a, j)$$

On introduit pour chaque temps  $t$ , le coût  $r_t(x, a) \in \mathbb{R}$  où  $x \in S$  et  $a \in A$ . Notre but est de choisir une suite  $a_1, \dots, a_n$  d'actions afin de minimiser le coût global. On appelle *stratégie* du contrôleur la séquence  $\pi = (d_1, \dots, d_t, \dots)$  où  $d_i$  est la fonction de décision au temps  $i$ , c'est à dire  $d_i : E \rightarrow \mathcal{A}$  où  $E$  caractérise la classe de la stratégie.

### Exemples de classes de stratégies :

1. (Boucle ouverte)  $E = \emptyset$ , la décision peut être différente à chaque instant mais elle ne dépend ni de l'état du système ni de son passé
2. (Markov déterministe - MD)  $d_i : S \rightarrow \mathcal{A}$ , la décision dépend à chaque instant uniquement de l'état présent.
3. (Markov randomisé - MR)  $d_i : S \rightarrow \mathcal{P}(A)$ , le résultat est une distribution sur  $\mathcal{A}$ .
4. (History Deterministic - HD)  $d_i : H_i \rightarrow \mathcal{A}$  où  $H_i = (X_0, A_1, \dots, X_{i-1}, A_{i-1}, X_i)$
5. (History Randomized - HR)  $d_i(H_i) \in \mathcal{P}(A)$

On va montrer que la 2<sup>e</sup>, 3<sup>e</sup> et 4<sup>e</sup> classes sont en fait aussi "puissantes" mais que la première est strictement moins forte que la seconde.

*Remarque 1.0.1.* Dans le cas des classes de stratégie randomisée, on aurait aussi pu prendre  $E = S \times \mathcal{E}$  où  $\mathcal{E}$  est un "bruit" et on aurait alors  $d_i : S \times \mathcal{E} \rightarrow A$  plutôt que  $\mathcal{P}(A)$ .

**Un exemple :** On considère un jeu à deux joueurs auquel on a beaucoup joué. On a deux stratégies une défensive et une offensive. La stratégie défensive permet de gagner avec probabilité  $p_1 = 0.1$ , faire ex-aequo avec probabilité  $p_0 = 0.6$  et perdre avec probabilité  $p_{-1} = 0.3$ . La stratégie offensive permet de gagner avec probabilité  $q_1 = 0.45$  et perdre avec probabilité  $q_{-1} = 0.55$ .

On fait maintenant un tournoi à deux manches où on gagne le tournoi si on a plus de victoire que de défaites. Regardons le comportement des différentes classes de stratégies :

- ▷ Avec une stratégie en boucle ouverte, on a quatre décisions possibles  $dd, do, od, oo$ . Le gain pour  $dd$  est  $(p_1^2 + 2p_1p_0) - (p_{-1}^2 + 2p_{-1}p_0)$ ; en calculant l'espérance du gain, on obtiendrait quelque chose de négatif.
- ▷ On considère maintenant la stratégie en boucle fermée suivante (MD) : jouer offensif puis si j'ai gagné jouer défensif sinon jouer offensif. En traçant un arbre des possibilités, on obtient comme gain avec cette stratégie  $q_1(p_1 + p_0) - q_{-1}^2 > 0$

Conclusion, la classe des stratégies en boucle ouverte est strictement moins efficace que la classe MD.

## 2 Horizon fini

On pose  $T$  l'horizon du système, c'est à dire que le système s'arrête au temps  $T$ . On note  $r_T(x) \in \mathbb{R}$  le coût à l'instant  $T$ . On veut optimiser

$$v^\pi(x) \stackrel{\text{def}}{=} \mathbb{E}_x^\pi \left( \sum_{t=1}^T r_t(X_t, A_t) \right)$$

où  $\pi$  est la stratégie,  $v$  la valeur,  $X_1 = x$  la valeur initiale,  $X_{t+1} = \varphi(X_t, A_t, w)$ ,  $A_t = d_t(H_t, w)$  ( $w$  randomization).

**Valeur optimale :**

$$v^*(x) = \sup_{\pi \in \{HR, HD, MR, MD\}} v^\pi(x)$$

$\pi \in \{HR, HD, MR, MD\}$  est  $\epsilon$ -optimal si  $\forall x, v^\pi(x) \geq v^*(x) - \epsilon$

**Lemme 2.1.** *Il existe une stratégie  $\epsilon$ -optimale pour tout  $\epsilon$*

$\pi^*$  est une stratégie optimale si  $\forall x \in S, v^*(x) = v^{\pi^*}(x)$  (une telle stratégie n'existe pas forcément).

**Programmation dynamique :** On note  $u_t^\pi(x)$  l'espérance du coût de  $t$  à  $T$  si  $X_t = x$  et  $u^*(x) = \sup_{\pi \in HR} u_t^\pi(x)$ .

$$\begin{aligned} u_T^\pi(x) &= r_T(x) = u_T^* \\ u_{T-1}^\pi(x) &= r_{T-1}(x, d_{T-1}(x)) + \sum_{y \in S} r_T(y) \cdot P(x, d_{T-1}(x), y) \\ u_t^\pi(x) &= r_t(x, d_t(x)) + \sum_y P(x, d_t(x), y) \cdot u_{t+1}^{\pi'}(y) \end{aligned}$$

où  $\pi'$  a  $(x, d_t(x), y)$  dans son historique.

**Principe d'optimalité :** Construisons une nouvelle fonction

$$w_t(x) = \sup_{a \in \mathcal{A}} \left( r_t(x, a) + \sum_y P(x, a, y) \cdot w_{t+1}(y) \right) \quad w_T(x) = r_T(x)$$

**Théorème 2.1.** *Si  $\pi \in HR$  alors,  $v^*(x) = w_1(x)$*

**Démonstration :** Commençons par montrer que  $w_1(x) \geq v^\pi(x) \forall \pi \in HR$ . On compare  $u_t^\pi(x)$  et  $w_t(x)$  :

- ▷  $w_T(x) \geq u_T^\pi(x)$  (car égaux)
- ▷ Supposons que  $\forall x w_t(x) \geq u_t^\pi(x)$  alors

$$\begin{aligned}
u_{t-1}^\pi(x) &= r_t(x, d_{t-1}(x)) + \sum_y P(x, d_{t-1}(x), y) u_t^\pi \\
&\leq r_{t-1}(x, d_{t-1}(x)) + \sum_y P(x, d_{t-1}(x), y) w_t(x) \\
&\leq \sup_a \left( r_t(x, a) + \sum_y P(x, a, y) w_t(a) \right) \\
&= w_{t-1}(x)
\end{aligned}$$

Pour finir la preuve, il faut construire une stratégie  $\epsilon$ -optimale qui soit dans MD. Supposons que nous ayons déjà fabriqué une telle stratégie pour  $t \rightarrow T$ , alors à l'étape  $t - 1$  on choisit  $d_{t-1}(x)$  tel que  $u_{t-1}^\pi \geq w(x) - (T-t+1)\epsilon$  ce qui signifie que  $u_1^\pi(x) \geq w(x) - T\epsilon$  c'est à dire  $\sup_{\pi \in MD} v^\pi(x) = \sup_{\pi \in MD} u_1^\pi(x) \geq w(x)$ .

En fait, nous avons même montré que

$$\sup_{\pi \in HR} v^\pi = \sup_{\pi \in MD} v^\pi = w$$

**Exemple :** On considère le problème du temps d'arrêt optimal pour une chaîne de Markov. Plus précisément on considère  $N$  candidats à un poste (de secrétaire par exemple) que l'on interview un par un. Après chaque candidat, on doit décider si on embauche ou pas le candidat. Si on embauche le candidat, tous les suivants rentrent chez eux, et si on ne l'embauche pas le candidat rentre chez lui. On modélise de la manière suivante :  $N$  correspond à notre horizon fini, l'ensemble des états est  $\{0, 1, \Delta\}$  où 0 signifie que le candidat courant n'est pas le meilleur parmi ceux que l'on a déjà vu, 1 signifie qu'il s'agit effectivement du meilleur parmi ceux que l'on a vu et  $\Delta$  est l'état bloquant (on ne test pas ce candidat parce qu'on a déjà choisi un candidat). L'ensemble des actions est  $\{\text{continue}(c), \text{stop}(s)\}$ .

$$\begin{array}{lll}
r_N(0) = 0 & r_N(\Delta) = 0 & r_N(1) = 1 \\
r_t(0, c) = 0 & r_t(1, c) = 0 & r_t(0, s) = 0
\end{array}$$

$$r_t(1, s) = \text{probabilité que le meilleur soit parmi les } t \text{ premiers} = \frac{t}{N}$$

$$\begin{array}{lll}
P(\cdot, s, \Delta) = 1 & P(0, c, 0) = 1 - \frac{1}{t+1} & P(0, c, 1) = \frac{1}{t+1} \\
P(1, c, 1) = \frac{1}{t+1} & P(1, c, 0) = 1 - \frac{1}{t+1} & \\
w_N(0) = 0 & w_N(1) = 1 & w_N(\Delta) = 0
\end{array}$$

$$\begin{aligned}
w_{t-1}(0) &= \frac{1}{t} w_t(1) + \frac{t-1}{t} w_t(0) \\
w_{t-1}(1) &= \max \left( \frac{t}{N}, \frac{1}{t} w_t(1) + \frac{t-1}{t} w_t(0) \right) \\
&= \max \left( \frac{t}{N}, w_{t-1}(0) \right)
\end{aligned}$$

Supposons que  $\forall t w_t(1) = w_t(0)$ , alors les équations précédentes nous donne une contradiction. Il existe un temps  $\tau$  tel que  $w_{\tau-1}(1) = \frac{\tau}{N}$ , et on considère le  $\tau$  maximal vérifiant cette propriété. On peut vérifier qu'avant  $\tau$  le coût optimal est de ne pas s'arrêter et après  $\tau$  le maximum est atteint et on reste donc constant. Ce meilleur  $\tau$  vérifie une équation dont la solution est de l'ordre de  $\frac{N}{e}$ .

**Exemples :**

- ▷ Maniot à plusieurs bras : consiste en  $K$  chaînes de Markov, la décision devant choisir quelle machine activer.
- ▷ Inventaire : on a un stock de capacité  $C$ . La meilleur stratégie est la stratégie  $s - S$  où  $s < S \leq C$ . Lorsque le stock descend en dessous de  $s$ , on le re-remplit jusqu'à  $S$ .
- ▷ Contrôle des queues : on a un serveur devant traiter des requêtes arrivant selon un processus de poisson de paramètre  $\lambda$  et capable d'en traiter  $\mu$  par unité de temps. Les requêtes attendent donc dans une queue avant d'être traitées. On place un contrôleur qui accepte ou rejette les requêtes avant la queue. Il y a une pénalité en cas de rejet et aussi en cas d'attente trop longue dans la queue.
- ▷ Le problème précédent se généralise au cas de  $n$  queues avec un seul contrôleur. Si toutes les queues se vident au même rythme, il suffit de placer les paquets dans la queue la moins remplies.

**Quand est-ce qu'il existe des stratégies optimales ?** Cela revient à se demander quand le "sup" est atteint dans l'équation d'optimalité :

$$w_t(x) = \sup_{a \in \mathcal{A}} \left( r_t(x, a) + \sum_y P(x, a, y) w_{t+1}(y) \right)$$

1.  $\mathcal{A}$  est un ensemble fini (cas fréquent dans les exemples précédent)
2.  $\mathcal{A}$  est compact,  $r_t(x, \cdot)$  et  $P(x, \cdot, y)$  sont continues en  $a$
3.  $\mathcal{A}$  est compact,  $r$  est semi-continue inférieurement et positive et  $P$  est semi-continue supérieurement.

### 3 Coûts discounted dans le cadre horizon infini

On prend des hypothèses supplémentaires par rapport au cas fini :

1. La récompense  $r(s, a)$  ne dépend pas du temps  $t$  (on est homogène par rapport au temps)
2.  $P(i, a, j)$  ne dépend pas de  $t$
3.  $|r(s, a)| < M$  (c'est bien sur le cas si l'ensemble des états est fini)
4. Le système a un facteur de discount  $\lambda \in [0; 1]$
5. Si le contrôleur emploie une stratégie  $\pi = (d_1, d_2, \dots, d_n, \dots)$ , alors la valeur de  $\pi$  est

$$v^\pi(s) = \mathbb{E}_s^\pi \left( \sum_{t=1}^{\infty} \lambda^{t-1} r(X_t, A_t) \right) \leq \frac{M}{1-\lambda}$$

#### 3.1 Notations vectorielles

L'espace des états étant supposé dénombrable, on peut voir  $v^\pi$  comme un vecteur et obtenir l'équation (avec  $\pi \in HD$ ) :

$$v^\pi(s) = r(s, d_1(s)) + \lambda \sum_j P(s, d_1(s), j) v^{\pi'}(j) \quad \text{où } \pi' = (d_2, d_3, \dots)$$

$$v^\pi = r_{d_1} + \lambda P_{d_1} v^{\pi'}$$

où  $r_d = (r(s, d(s)))_{s \in S}$  et  $P_d = (P(i, d(i), j))_{i, j \in S}$ .

Dans le cas particulier où notre stratégie ne dépend pas du temps,  $\pi = (d, d, d, \dots)$  on obtient l'équation suivante :

$$\begin{aligned} v^\pi &= r_d + \lambda P_d v^\pi \\ v^\pi &= (I - \lambda P_d)^{-1} r_d \end{aligned}$$

avec  $P_d$  stochastique donc  $\lambda P_d$  est sous-stochastique et  $(I - \lambda P_d)^{-1}$  est inversible avec

$$(I - \lambda P_d)^{-1} = \sum_{k \in \mathbb{N}} (\lambda P_d)^k$$

### 3.2 Équation d'optimalité

Posons

$$w(s) = \sup_a \left( r(s, a) + \lambda \sum_{j \in S} P(s, a, j) w(j) \right)$$

c'est à dire vectoriellement :

$$w = \sup_a (r_a + \lambda P_a w)$$

ce que l'on nomme l'équation de point fixe de Bellman puisqu'en introduisant l'opérateur  $\mathcal{L} : u \mapsto \sup_a (r_a + \lambda P_a u)$ , l'équation se réécrit  $w = \mathcal{L}w$ .

**Théorème 3.1** (Hann-Banach). *Dans un espace de Banach  $\mathcal{V}$ , si l'opérateur  $T : \mathcal{V} \rightarrow \mathcal{V}$  vérifie*

$$\|Tu - Tv\| \leq c \|u - v\|$$

avec  $c < 1$  alors  $T$  a un point fixe  $u^*$  et  $u^* = \lim_{n \rightarrow +\infty} T^n u_0$

**Démonstration :** en exercice (montrer que la suite des itérés de  $u_0$  est de Cauchy).

**$\mathcal{L}$  est contractante :** Notons  $a_v$  la meilleure action pour  $v$  (à  $\epsilon$  près).

$$\begin{aligned} r(s, a_v) + \lambda P(s, a_v, j)v(j) &\geq (\mathcal{L}v)(s) - \epsilon \\ \mathcal{L}u(s) - \mathcal{L}v(s) &\leq r(s, a_u) + \lambda \sum_j P(s, a_u, j)u(j) - r(s, a_v) - \lambda \sum_j P(s, a_v, j)v(j) + \epsilon \\ &\leq \lambda \sum_j P(s, a_u, j)(u(j) - v(j)) + \epsilon \\ &\leq \sum_j P(s, a_u, j)\|u - v\| + \epsilon \\ &\leq \lambda \|u - v\| + \epsilon \end{aligned}$$

**Théorème 3.2.** Soit  $w$  la solution de l'équation de Bellman ( $w = \mathcal{L}w$ ), alors

$$\begin{aligned}
w &= \sup_{\pi \in HR} v^\pi \\
&= \sup_{\pi \in HD} v^\pi \\
&= \sup_{\pi \in MR} v^\pi \\
&= \sup_{\pi \in MD} v^\pi \\
&= \sup_{\substack{\pi=(d,d,d,\dots) \\ d \text{ déterministe}}} v^\pi
\end{aligned}$$

**Démonstration :** Les inclusions  $\{(d, d, d, \dots) \mid d \text{ déterministe}\} \subset MD \subset MR \subset HD \subset HR$  nous donne les inégalités  $\geq$  sauf la première.

Réciproquement, prenons n'importe quelle stratégie  $(d_1, \dots, d_n)$  :

$$\begin{aligned}
w &\geq r_{d_1} + \lambda P_{d_1} w \\
&\geq r_{d_1} + \lambda P_{d_1} (r_{d_2} + \lambda P_{d_2} w) \\
&\geq \dots \\
w &\geq \lambda^n P_\pi^n w - \sum_{k=n}^{\infty} \lambda^k P_\pi^k r_{d_{k+1}}
\end{aligned}$$

où  $P_\pi^k = P_{d_1} P_{d_2} \dots P_{d_n}$ . Pour  $n$  suffisamment grand, le second terme est positif et  $w \geq v^\pi$  et bien sur il existe  $d$  tel que  $\pi = (d, d, d, \dots)$  est  $\epsilon$ -optimale (Il suffit de choisir  $d$  vérifiant  $r_d + \lambda P_d w \geq w - \epsilon$ ).

Posons à présent  $L_d : u \mapsto r_d + \lambda P_d u$ .

**Théorème 3.3.** Si  $\mathcal{A}$  est fini et  $S$  compact, alors le supremum est un maximum et une stratégie optimale existe.

Remarquons que si  $u \leq v$ , alors  $\mathcal{L}u \leq \mathcal{L}v$ ,  $L_d u \leq L_d v$  et  $\mathcal{L}(v + c\mathbb{1}) = \mathcal{L}v + \lambda c\mathbb{1}$  et de même pour  $d$ .  
Remarquons aussi que le point fixe est unique mais pas la stratégie optimale.

### 3.3 Algorithme par itération de valeur (VIA)

Supposons à présent qu'une telle stratégie optimale existe.

**Algorithme :**

```

 $v^0 \leftarrow$  arbitraire
repeat
   $v^{n+1} \leftarrow \mathcal{L}v^n$ 
until  $\|v^{n+1} - v^n\| \leq \frac{\epsilon^{(1-\lambda)}}{2\lambda}$ 

```

**Théorème 3.4.** L'algorithme termine en  $N$  étapes et la stratégie  $\pi = (d, d, d, \dots)$  avec  $d \in \arg \max(\mathcal{L}v^N)$  est  $\epsilon$ -optimale.

**Démonstration :** Avec  $v^d$  la valeur associée à la stratégie  $\pi = (d, d, d, \dots)$  et  $v^*$  la valeur optimale :

$$\begin{aligned}
\|v^d - v^*\| &\leq \|v^d - v^N\| + \|v^N - v^*\| \\
\|v^d - v^N\| &\leq \|v^d - L_d v^N\| + \|L_d v^N - v^N\| \\
&= \|L_d v^d - L_d v^N\| + \|\mathcal{L}v^N - \mathcal{L}v^{N-1}\| \\
&\leq \lambda \|v^d - v^N\| + \lambda \|v^N - v^{N-1}\| \\
\|v^d - v^N\| &\leq \frac{\lambda}{1-\lambda} \|v^N - v^{N-1}\| \\
\|v^N - v^*\| &\leq \frac{\lambda}{1-\lambda} \|v^N - v^{N-1}\|
\end{aligned}$$

où la seconde inégalité est obtenu avec un calcul similaire. On a donc

$$\|v_d - v^*\| \leq \epsilon$$

*Remarque 3.3.1.* D'après le théorème de Hann-Banch, la convergence est exponentielle :

$$\|v^n - v^*\| \leq \frac{\lambda^n}{1-\lambda} \|v^1 - v^0\|$$

### 3.4 Algorithme par itération de stratégie (PIA)

$d_0 \leftarrow$  stratégie arbitraire

**repeat**

$v^n \leftarrow (I - \lambda P_{d_n})^{-1} r_{d_n}$  la valeur de  $d_n$

$d_{n+1}(s) \in \arg \max_a \left( r(s, a) + \lambda \sum_j P(s, a, j) v^n(j) \right)$

**until**  $d_{n+1} = d_n$

**Théorème 3.5.**  $\triangleright$  L'algorithme PIA converge vers une solution optimale.

$\triangleright$  Quand  $S$  et  $A$  sont finis, la convergence se déroule en temps fini.

**Démonstration :** (idée de la preuve)  $v^n$  est croissant et borné par  $\frac{M}{1-\lambda}$

*Remarque 3.4.1.* La convergence est bien plus rapide que dans l'algorithme VIA.

**Versión modifiée de PIA** À la 3<sup>ème</sup> ligne de l'algorithme précédent, on calcule la valeur  $v^n$  en inversant une matrice, opération assez couteuse dont on voudrait se débarrasser. On utilise le fait que  $v^n = L_{d_n} v^n$  :

$d_0 \leftarrow$  stratégie arbitraire

**repeat**

$v^n \leftarrow L_{d_n}^k v^{n-1}$

$d_{n+1} \in \arg \max_d (r_d + \lambda P_d v^n)$

**until**  $d_{n+1} = d_n$

### 3.5 Exemple

On a des paquets qui arrivent dans une file avec le débit  $\lambda$  qui sont traités par un serveur au débit  $\mu$ . On dirige un contrôleur qui contrôle l'admission dans la pile.

Si les débits (éléments de  $Q$ ) sont tous bornés, alors la chaîne de Markov à temps continu (CTMC) peut être transformée en une chaîne de Markov à temps discret (DTMC) avec une matrice de transition  $P$  en effectuant une uniformisation.

Les processus d'entrée et de sortie sont des processus de Poisson. On uniformise à l'aide d'un processus de Poisson de débit  $K$  plus grand que tous la somme des autres débits. Le principe est que l'on peut modéliser

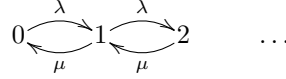


les processus de Poisson dont les débits sont plus faible dans ce nouveau débit. En effet, si on a un processus de Poisson de débit  $\lambda$  et que l'on sélectionne un point avec probabilité  $p$  alors on obtient un processus de Poisson avec débit  $p\lambda$ .

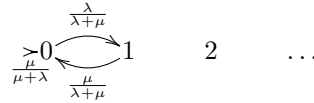
On pose  $P_{ij} = (I_{ij} + \frac{Q_{ij}}{K})$  ce qui nous donne les probabilités de la chaîne discrète. La distribution stationnaire est la même

$$\pi P = \pi \left( I + \frac{Q}{K} \right) = \pi \iff \pi Q = 0$$

Dans notre cas on a pour  $Q$  :



et donc avec  $K = \lambda + \mu$  :



La récompense va être ici un coût. Le rejet d'un paquet coûte  $R$ . Si on veut minimizer le temps moyen de réponse des paquets alors le coût instantané de garder  $n$  paquets dans la file est  $n$ .

On commence la modélisation :  $S = \mathbb{N}$  est le nombre de paquets dans la file,  $\mathcal{A} = \{\text{accept, reject}\}$ .

$$v(x) = \frac{\mu}{\lambda + \mu} (x + \alpha v(x - 1)) + \frac{\lambda}{\lambda + \mu} \min(x + \alpha v(x + 1), x + R + \alpha v(x))$$

Pour faciliter la programmation, on va supposer que la file a une capacité finie  $C$  (et donc  $S = \mathbb{N}_C$ ).

*Exercice 3.5.1.* Pour le prochain cours, implémenter VIA, PIA et MPIA et tester lequel est le plus rapide. Décrire la stratégie optimale. Que se passe-t-il lorsque  $C \rightarrow +\infty$ ? Le prouver.

### 3.6 Stratégies structurées

Une stratégie  $\pi = (d, d, d, \dots)$  est monotone si les espaces d'états et d'action sont ordonnés (on considèrera ici que ce sont des parties de  $\mathbb{N}$ ) et si  $x \geq x' \implies d(x) \geq d(x')$ . Supposons que l'on est  $V^I \subset V$  et  $D^I \subset D$  des parties stable par l'opérateur de Bellman  $\mathcal{L}$  de la manière suivante :

- (c<sub>1</sub>)  $v \in V^I \implies \mathcal{L}v \in V^I$
- (c<sub>2</sub>)  $v \in V^I \implies \exists d \in D^I \cap \arg \max_d (L_d v)$
- (c<sub>3</sub>)  $V^I$  est fermé

**Théorème 3.6.** *Sous ces hypothèses, il existe une stratégie optimale dans  $D^I$ .*

**Démonstration :** Utilisons l'algorithme par itération de valeur à partir de  $v_0 \in V^I$ .  $\lim_{n \rightarrow \infty} v_n = v^*$  et  $v^* \in V^I$  avec  $d^* \in \arg \max_d (L_d v^*) \cap D^I$  une stratégie optimale.

Dans la suite, on prend :

- ▷  $V^I$  est l'ensemble des fonctions de valeur monotônes
- ▷  $D^I$  est l'ensemble des stratégies monotônes

**Théorème 3.7.** *Si*

- ▷  $r(s, a)$  est croissante par rapport à  $s$
- ▷  $\sum_{j \geq k} p(s, a, j)$  est croissant par rapport à  $s$
- ▷  $r(s, a)$  est supermodulaire en  $s$  et  $a$
- ▷  $\sum_{j \geq k} p(s, a, j)$  est supermodulaire en  $s$  et  $a$

Alors  $V^I$  et  $D^I$  satisfont  $c_1, c_2$  et  $c_3$  et il existe une stratégie optimale qui est monotone.

**Définition 3.1** (Fonction supermodulaire). Soit  $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ .  $f$  est supermodulaire si

$$\forall x^+ \geq x^- \text{ et } y^+ \geq y^-, f(x^+, y^+) + f(x^-, y^-) \geq f(x^+, y^-) + f(x^-, y^+)$$

*Exercice 3.6.1.*

▷ Si  $f \in \mathcal{C}^2$ ,  $\frac{\partial^2 f}{\partial x \partial y} \geq 0 \implies f$  est supermodulaire.

▷ Si  $f(x, y) = h(x + y)$  avec  $h$  convexe croissante alors  $f$  est supermodulaire.

**Proposition 3.1.** Si  $g$  est supermodulaire, posons  $f = \max \{y \in \arg \max_y (g(x, y))\}$  alors  $f$  est croissante.

**Démonstration :** Soit  $x^+ \geq x^-$ , prenons  $y \leq f(x^-)$ . Par définition de  $f$ ,  $g(x^-, f(x^-)) \geq g(x^-, y)$  (i) alors par supermodularité de  $g$

$$\begin{aligned} g(x^+, f(x^-)) + g(x^-, y) &\geq g(x^+, y) + g(x^-, f(x^-)) \\ g(x^+, f(x^-)) &\geq g(x^+, y) \quad \text{grâce à (i)} \end{aligned}$$

Donc il existe un élément  $f(x^+) \in \arg \max g(x^+, \cdot)$  supérieur à  $f(x^-)$ .

**Démonstration :** (Du théorème) Si  $v_n \in V^I$ , posons  $v_{n+1} = \mathcal{L}v_n$  et pour  $s' \geq s$  :

$$\begin{aligned} v_{n+1}(s) &= \max_a \left( r(s, a) + \lambda \sum_j P(s, a, j) v_n(j) \right) \\ &\leq \max_a \left( r(s', a) + \lambda \sum_j P(s, a, j) v_n(j) \right) \\ &\leq \max_a \left( r(s', a) + \lambda \sum_j P(s', a, j) v_n(j) = v_{n+1}(s') \right) \end{aligned}$$

Par les propriétés des fonctions supermodulaires,  $w(s, a) \stackrel{\text{def}}{=} r(s, a) + \sum_j P(s, a, j) v_n(j)$  est une fonction supermodulaire en  $s$  et  $a$ . En utilisant, la propriété pour  $w(s, a)$ , la stratégie est croissante en  $s$ .

**Exemple :** Problème du contrôle d'admission

La stratégie optimale avec la propriété de monotonie est :  $\exists \theta^*$  tel que  $\forall x \leq \theta^*, d^*(x) = 0$  et  $\forall x > \theta^*, d^*(x) = 1$

## 4 Récompense moyenne

On veut s'affranchir de la dépendance sur le facteur de discount. On se place donc dans un nouveau cadre :

- ▷  $P$  et  $r$  sont indépendant du temps
- ▷  $|r(s, a)| < M, \forall s, a$
- ▷ L'espace des états est fini

### 4.1 Critère d'optimalité

$$v_N^\pi(s) = \mathbb{E}_s^\pi \left[ \sum_{t=1}^N r(X_t, A_t) \right]$$

$$g^\pi(s) = \lim_{N \rightarrow \infty} \frac{1}{N} v_N^\pi(s)$$

Si le gain  $g$  n'existe pas (pas de limite), on considère  $g^+$  et  $g^-$  respectivement la limsup et liminf de  $(\frac{1}{N} v_N^\pi(s))_N$ .  $\pi^*$  est optimal en moyenne si  $g^{\pi^*}(s)$  existe et que  $\forall \pi \in HR \forall s, g^{\pi^*} \geq g^{+\pi}(s)$ .

Par définition du gain, il est facile de voir que on peut se contenter de ne considérer que des stratégies markoviennes.

## 4.2 Processus de récompense Markovien (MRP)

On fixe une stratégie  $\pi = (d, d, d, \dots)$ . Le système devient alors une chaîne de Markov avec pour matrice de transition  $P(i, j) = P(i, d(i), j)$ . Le gain d'un MRP est

$$g^{d^*}(s) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N \sum_j P^{t-1}(s, j) r(j)$$

$$g = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N P^{t-1} r$$

Si notre chaîne de Markov est apériodique et irréductible, on a

$$P^* = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N P^{t-1} \rightarrow \mathbb{1} \nu$$

où  $\mathbb{1}$  est un vecteur contenant que des 1 et  $\nu$  est la distribution stationnaire qui existe toujours sous ces hypothèses. Dans ce cas  $g(s) = P^* r$  est constant et vaut  $\sum_i \nu_i r(i)$ .

En dehors des hypothèses d'apériodicité et d'irréductibilité,  $P^*$  existe encore pour une chaîne de Markov. Si  $P$  à  $k$  classes finales :

$$\begin{bmatrix} P_1 & 0 & \dots & 0 & 0 \\ 0 & P_2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & P_k & 0 \\ Q_1 & Q_2 & \dots & Q_k & P_{k+1} \end{bmatrix}$$

$$\begin{matrix} P1 & 0 & \dots & 0 & 0 \\ 0 & P2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & Pk & 0 \\ Q1 & Q2 & \dots & Qk & P(k+1) \end{matrix}$$

$$P \rightarrow P^*$$

$$\begin{matrix} P1^* & 0 & \dots & 0 & 0 \\ 0 & P2^* & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & Pk^* & 0 \\ Q1' & Q2' & \dots & Qk' & 0 \end{matrix}$$

$$\text{avec } Q'_i = (I - P_{k+1})^{-1} Q_i P_i^*$$

**Définition 4.1** (Biais). Le vecteur de biais  $h$  est défini par

$$h = Hr$$

$$H = (I - P + P^*)^{-1} (I - P^*)$$

Si la chaîne est apériodique alors

$$\begin{aligned} h &\stackrel{\text{admis}}{=} \sum_{t=0}^{\infty} (P^t - P^*) r \\ &= \sum_{t=0}^{\infty} P^t (r - g) \end{aligned}$$

car  $PP^* = P^*$  donc  $h = \lim_{N \rightarrow \infty} \left( \sum_{t=0}^N P^t r - Ng \right) = v_N$

$$v_N = Ng + h + o(1)$$

### 4.3 Équation d'évolution

Soit  $(g, h)$  le gain et le biais d'un MRP, alors :

1.  $(I - P)g = 0$
2.  $g + (I - P)h = r$

Réciproquement, si  $(g, h)$  satisfait (1) et (2) alors il s'agit du gain et du biais (à une addition avec un vecteur  $u$  vérifiant  $(I - P)u = 0$  près). Si on voulait aussi obtenir  $h$ , il faudrait ajouter l'équation  $P^*h = 0$ .

Dans le cas où le MRP est irréductible, la première équation est inutile et  $(g, h)$  est donné par  $g$  est alors un scalaire)

$$g \cdot \mathbf{1} + (I - P)h = r$$

nous donne à la fois  $g$  et  $h$  (à un vecteur constant près).

### 4.4 Classification des Processus de décision Markovien (MDP)

**Définition 4.2** (Unichaîne). Un MDP est unichaîne si  $\forall d$ , le MRP associé est unichaîne (avec la même classe finale). (Une chaîne de Markov est unichaîne si elle a une seule classe finale avec éventuellement des états transients).

**Définition 4.3** (Communiquant). Un MDP est communiquant si il existe une stratégie  $(d, d, d, \dots)$  tel que  $j$  est atteignable depuis  $i$  dans le MRP.

**Définition 4.4** (Faiblement communiquant). Il existe un ensemble d'états qui sont communiquant et un ensemble d'états qui sont transient pour chaque stratégies.

*Exercice 4.4.1.* Trouvez une MDP (2 états, 2 stratégies) qui soit seulement soit unichaîne, soit communiquant, soit faiblement communiquant.

**Théorème 4.1.** Si on a un MDP faiblement communiquant alors, pour toute stratégie  $d$  (avec gain associé  $g^d$ ), il existe une stratégie  $\delta$  unichaîne telle que  $g^\delta \geq g^d$ . Par conséquent, il existe une stratégie optimale qui est unichaîne.

Le principe de la preuve est de fabriquer une stratégie  $\delta$  qui va utiliser le caractère faiblement communiquant pour amener toute la chaîne dans l'état le plus "rentable" pour la stratégie  $d$ .

### 4.5 Équation d'optimalité pour les unichaînes

Soit  $d$  unichaîne, alors  $f, g$  et  $h$  sont donnés par  $g\mathbf{1} + (I - P)h - r = 0$ . L'équation d'optimalité est alors :

$$\max_d (-g\mathbf{1} - (I - P_d)h + r_d) = 0$$

Écrivons-la sous la forme scalaire suivante :

$$g + h(s) = \max_a \left( r(s, a) + \sum_j P(s, a, j)h(j) \right)$$

**Justification heuristique de l'équation d'optimalité :**

$$\begin{aligned}
v_N &= Ng + h + o(1) \\
\rightsquigarrow v_N^* &= Ng^* + h^* + o(1) \\
v_{N+1}^* &= \max_d (r_d + P_d v_N^*) \\
&= \max_d (r_d + P_d (Ng^* + h^* + o(1))) \\
(N+1)g^* + h^* + o(1) &= \max_d (r_d + Ng^* + P_d h^* + o(1)) \\
\max_d (r_d - g^* + (P_d - I)h^*) &= 0
\end{aligned}$$

**Théorème 4.2.** Soit un MDP unichaine et  $(g^*, h^*)$  les solutions de l'équation d'optimalité alors  $g^*$  est le gain optimal et la stratégie  $d^*$  définie par

$$d^* \in \arg \max_d (r_d + P_d h^*)$$

est optimal en moyenne.

**Démonstration :**  $g^*$  est supérieur au gain de n'importe quelle stratégie et c'est le gain de la stratégie  $d^*$ .  
On définit l'opérateur  $L$  par

$$\begin{aligned}
L_d v &= r_d + P_d v \\
L v &= \max_d (r_d + P_d v)
\end{aligned}$$

**Algorithme par itération de valeur :**

$v^0 \leftarrow$  arbitraire  
**repeat**  
 $v^{n+1} \leftarrow L v^n$   
**until**  $v^{n+1}(i) - v^n(i)$  ne dépend plus de  $i$  à  $\epsilon$  près  
choisir  $d \leftarrow \arg \max_d L_d v^{n+1}$   
La condition d'arrêt dans la boucle peut se traduire par

$$\max_i (v^{n+1}(i) - v^n(i)) - \min_i (v^{n+1}(i) - v^n(i)) \leq \epsilon$$

**Algorithme par itération de stratégie :**

choisir  $d^0 \leftarrow$  arbitraire  
**repeat**  
résoudre  $g_n \mathbb{1} + (I - P_{d_n})h = r$   $d_{n+1} \leftarrow \arg \max_d (r_{d_n} + P_{d_n} h_n)$  (en conservant  $d_n$  si possible)  
**until**  $d_{n+1} = d_n$

Deuxième partie

# Réseaux de requêtes Markoviennes - Product Forms

(jean.mairesse@liafa...)

## 5 Model de requêtes en temps discret

On considère la situation suivante :

Notation de Kendall :  $\cdot / \cdot / 1 / \infty$  (Arrivées / / nombre de serveurs / taille de la pile)

**Comportement du modèle :** à chaque instant  $n$ , on a

–  $A_n$  clients qui arrivent,  $A_n \sim \mu$ ,  $(A_n)_n$  iid,  $p \in [0; 1]$

$$\mu : \begin{cases} \mu(0) = 1 - p \\ \mu(2) = p \end{cases}$$

– Un client est servi si la pile n'est pas vide.

**Equations :** Soit  $X_n$  le nombre de clients dans la pile au début de l'instant  $n$ .

$$\begin{aligned} X_{n+1} &= (X_n + A_n - 1)^+ & x^+ &= \max(x, 0) \\ (A_n)_n &\text{iid} & A_0 &\sim \mu \end{aligned}$$

$(X_n)_{n \in \mathbb{N}}$  est une chaîne de Markov avec  $\mathbb{N}$  comme espace d'état. La matrice de transition  $\mathbb{P}$  contient  $1 - p$  dans les cases  $(0, 0)$  et  $(i, i + 1)$  et  $p$  dans les cases  $(i - 1, i)$ ,  $\forall i \in \mathbb{N}$ .

**Cas 1** |  $p > \frac{1}{2}$ , instable

**Cas 2** |  $p < \frac{1}{2}$ , on a un équilibre

On se place dans le second cas et on cherche la solution stable. Soit  $X_*$  le nombre de clients dans le comportement à l'équilibre et  $X_* \sim \pi$ ,  $\pi(k) = P(X_* = k)$

$$\begin{cases} \pi(k) = \pi(k - 1)p + \pi(k + 1)(1 - p) & k > 0 \\ \pi(0) = \pi(0)(1 - p) + \pi(1)(1 - p) \\ \sum_k \pi(k) = 1 \end{cases}$$

Pour résoudre l'équation récurrente d'ordre 2, on suppose que  $\pi(k) = Cu^k$ , on substitue dans l'équation précédente pour obtenir  $(1 - p)u^2 - u + p = 0$  ce qui nous donne  $u = \frac{1 \pm \sqrt{1 - 4p(1 - p)}}{2(1 - p)}$  d'où  $u = 1$  ou  $\frac{p}{1 - p}$ . La seule solution possible pour que se soit sommable il faut avoir  $u = \frac{p}{1 - p} < 1 \iff p < \frac{1}{2}$ . Avec  $\sum_k \pi(k) = 1$ ,  $C = 1 - \frac{p}{1 - p}$  donc

$$\pi(k) = \left(1 - \frac{p}{1 - p}\right) \left(\frac{p}{1 - p}\right)^k \sim \text{Geom}\left(\frac{p}{1 - p}\right)$$

La distribution stable à l'équilibre est donné par l'équation  $\pi = \pi\mathbb{P}$ .

**Évaluation des performances :** Soit  $L$  le temps moyen passé par un client dans le système.

$$L = \sum_{k \in \mathbb{N}} (k + 1)\pi(k) = 1 + \sum_k k\pi(k)$$

Comme  $\pi = \text{Geom}\left(\frac{p}{1 - p}\right)$ , on a  $L = 1 + \frac{p}{1 - 2p}$  (exo).

## 6 Real-life network

**A) Packet switching (Internet)** Les messages sont découpés en paquets qui sont envoyés à travers le réseau de manière indépendante.

**B) Circuit switching** Une route est allouée à la transmission des données.

Modèles mathématiques :

**A)** réseaux de Jackson pour modéliser les packet switching network

**B)** réseaux à pertes (Loss network) pour modéliser les circuit switching network

On se place dans le cadre des processus de Markov à temps continu.

### 6.1 Processus Markovien

**Propriété de Markov** Soit  $(X_t)_t$  un processus sur l'espace d'état  $E$  au plus dénombrable.  $\forall t_1 < t_2 < t_3 \dots < t_{k+1}, x_1, x_2, \dots, x_{k+1} \in E$

$$\mathbb{P} [X_{t_{k+1}} = x_{k+1} | X_{t_1} = x_1, \dots, X_{t_k} = x_k] = \mathbb{P} [X_{t_{k+1}} = x_{k+1} | X_{t_k} = x_k]$$

Les transitions entre deux états vont toujours être représentées par une probabilité  $p_{ij}$  de passer de l'état  $i$  à l'état  $j$ , mais on va aussi avoir besoin de variables aléatoire suivant une loi  $\mathcal{Exp}(q_i)$  où  $q_i$  est propre à chaque état représentant le temps passé dans cet état.

**Loi exponentielle :** Soit  $\lambda \in ]0; +\infty[$ ,  $X \sim \mathcal{Exp}(\lambda)$  alors  $P(X \in A) = \int_A \lambda e^{-\lambda x} dx$ , c'est à dire  $P(X \geq t) = e^{-\lambda t}$

**Proposition 6.1** (Absence de mémoire).

$$P(X \geq t + s | X \geq t) = P(X \geq s)$$

En effet,

$$\frac{P(X \geq t + s)}{P(X \geq t)} = \frac{e^{-\lambda(t+s)}}{e^{-\lambda t}} = e^{-\lambda s}$$

**Lemme 6.1** (du réveil). Soit  $X_1 \sim \mathcal{Exp}(\lambda_1), X_2 \sim \mathcal{Exp}(\lambda_2)$ ,  $X_1$  indépendant de  $X_2$  et  $X = \min(X_1, X_2)$ . Alors,  $X \sim \mathcal{Exp}(\lambda_1 + \lambda_2)$  et  $P[X \in A, X = X_1] = P[X \in A]P[X = X_1] = P[X \in A] \left( \frac{\lambda_1}{\lambda_1 + \lambda_2} \right)$

Extension :  $(X_i)_{i \in \mathbb{N}}, X_i \sim \mathcal{Exp}(\lambda_i), \sum_i \lambda_i < +\infty, X = \min_{i \in \mathbb{N}} X_i, X \sim \mathcal{Exp}(\sum_i \lambda_i)$

### 6.2 Processus de Poisson

**Définition 6.1.** Un Processus de point sur  $\mathbb{R}^+$  est une séquence de variables aléatoires  $(T_i)_{i > 0}$  avec  $0 \leq T_1 < T_2 < \dots < +\infty$  et  $\lim_n T_n = +\infty$ . Le processus de comptage associé avec le processus de points est le processus  $N = (N_t)_{t \in \mathbb{R}^+}$  avec  $N_t = \sum_{k=1}^{+\infty} \mathbb{1}_{[0, t]}(T_k)$

**Théorème 6.1.** Considérons un processus de points tel que le processus de décompte  $N$  associé vérifie :

- $N_0 = 0$
- $\forall s, t \geq 0, N_{t+s} - N_s$  est indépendant des  $(N_u)_{u \leq t}$
- $\forall s, t, (N_{t+s} - N_t) \sim (N_t - N_0)$

Alors il existe  $\lambda \in ]0, +\infty[$  tel que les variables aléatoires  $T_1, T_2 - T_1, T_3 - T_2$  soient iid distribuées selon  $\mathcal{Exp}(\lambda)$ . On dit alors que le processus de points est un processus de Poisson (de paramètre  $\lambda$ ).

**Proposition 6.2.** Soit  $(T_n)_{n \in \mathbb{N}}$  un processus de Poisson de paramètre  $\lambda$ ,  $C$  un interval de  $\mathbb{R}^+$ . Posons  $N(C) = \sum_{k=1}^{\infty} \mathbb{1}_C(T_k)$ .  $N(C)$  est un processus de Poisson de paramètre  $\lambda \text{length}(C)$ .

Si  $X \sim \text{Poisson}(u)$ , alors  $P(X = k) = e^{-u} \frac{u^k}{k!}$

### 6.3 Processus de Markov

Soit  $(X_t)_{t \in \mathbb{R}^+}$  d'espace d'état  $E$ . On considère l'ensemble  $(T_i)_{i \in \mathbb{N}}$  des instants de sauts définis par :

$$\begin{cases} T_0 = 0 \\ T_{i+1} = \inf(t > T_i, X_t \neq X_{T_i}) \\ T_0 = 0 < T_1 < T_2 \end{cases}$$

Posons  $A = (A_{ij})_{i,j \in E}$ ,  $A_{ij} = P[X_{T_1} = j | X_0 = i]$ ,  $q_i = \frac{1}{\mathbb{E}_i(T_i)}$

**Théorème 6.2.**  $\triangleright (X_{T_n})_{n \in \mathbb{N}}$  est une chaîne de Markov de matrice de transition  $A$ .

$\triangleright P(X_{T_{n+1}} = j, T_{n+1} - T_n > t | X_{T_n} = i) = e^{-q_i t} A_{ij}$

Le processus de Markov est décrit par le couple  $(A, (q_i)_i)$ , ce qui est assez compliqué. On va essayer de tout encoder dans une seule matrice.

**Définition 6.2.** Un processus de Markov est dit

$\triangleright$  explosif si  $P(\lim_n T_n < +\infty) > 0$

$\triangleright$  non-explosif si  $P(\lim_n T_n = +\infty) = 1$

**Proposition 6.3.** Si  $\sup_i(q_i) < +\infty$  ou si  $A$  est récurrente alors le processus de Markov est non explosif.

### 6.4 Infinitesimal generator

On part d'un processus de Markov  $(X_t)_{t \in \mathbb{R}^+}$ . On note  $\mathbb{P}(t)_{i,j} = P(X_t = j | X_0 = i)$ . On pose ensuite

$$\mathbb{Q} = \lim_{t \rightarrow 0} \frac{\mathbb{P}(t) - Id}{t}$$

Mais,  $\forall i \neq j, \mathbb{P}(t)_{ij} = P(X_t = j | X_0 = i) = (q_i t) A_{ij} + o(t)$ , donc

$$\mathbb{Q}_{i,j} = q_i A_{i,j} \text{ et } \forall i, \mathbb{Q}_{i,i} = -q_i (\mathbb{P}(t)_{ii} = (1 - q_i)t + o(t))$$

**Définition 6.3.** Le générateur infinitésimal  $\mathbb{Q} = (\mathbb{Q}_{ij})_{i,j \in E}$  est donné par :

$$\mathbb{Q}_{ij} = \begin{cases} q_i A_{i,j} & i \neq j \\ -q_i & i = j \end{cases}$$

**Théorème 6.3.** Supposons que  $\sup_i(q_i) < +\infty$  alors

$$\mathbb{P}(t) = e^{t\mathbb{Q}} = \sum_{k=0}^{+\infty} \frac{t^k \mathbb{Q}^k}{k!}$$

$\mathbb{Q}$  joue le rôle de la matrice de transition en temps discret.

### 6.5 Du modèle au générateur infinitésimal

On considère le modèle présenté au début de cette partie.

Les clients arrivent selon un processus de poisson de paramètre  $\lambda$  et partent selon un processus de poisson de paramètre  $\mu$ .

En notation de Kendall :  $M/M/1/\infty$

Posons  $X_t$  le nombre de clients dans la pile ou le serveur au temps  $t (\in \mathbb{R}^+)$ .

**Proposition 6.4.**  $(X_t)$  est une processus de Markov de générateur infinitésimal

$$\mathbb{Q} : \begin{cases} \mathbb{Q}_{n,n+1} = \lambda \\ \mathbb{Q}_{n,n-1} = \mu \cdot \mathbb{1}_{\{n>0\}} \\ \mathbb{Q}_{n,n} = -(\lambda + \mu) \end{cases}$$



**Queue M/M/K/∞** Soit  $K > 1$ , on reprend la situation précédente avec cette fois  $K$  serveur.  
On a dans ce cas

$$Q : \begin{cases} Q_{n,n+1} = \lambda & \forall n \\ Q_{n,n-1} = n \cdot \mu & \forall 0 < n \leq K \\ Q_{n,n-1} = K \cdot \mu & \forall K < n \end{cases}$$

**Modèle d'Erlang (queue M/M/K/0)** Cette fois ci on a plus de pile et l'espace d'état est restreint à  $\{0, 1, \dots, K\}$ . Sur cet espace  $Q$  est la restriction du cas précédent.

rattraper  
le cours du  
21/01

**Réversibilité** Soit  $(X_t)_t$  un processus de Markov de générateur  $Q$  et de mesure invariante  $\pi$  avec  $X_0 \sim \pi$ .

**Proposition 6.5.** Soit  $T > 0$ ,  $(X_{T-t})_t$  est alors un processus de Markov de générateur  $\tilde{Q}$  donné par  $\forall i, j, \tilde{Q}_{i,j} = \frac{\pi(j)}{\pi(i)} Q_{j,i}$  et on a équivalence entre :

- ▷  $(X_{T-t})_t \sim (X_t)_t$
- ▷  $\forall i, j, \pi(i) Q_{ij} = \pi(j) Q_{ji}$

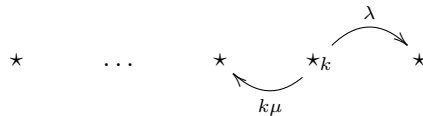
On dit alors que le processus de Markov est réversible.

**Théorème 6.4.** Soit  $(X_t)_t$  un processus de Markov de générateur  $Q$  et de probabilité invariante  $\pi$ . Si  $\exists \tilde{Q}$  ( $\forall i, j, \tilde{Q}_{ij} \geq 0, \tilde{Q}_{ii} = \sum_{j \neq i} \tilde{Q}_{ij}$ ) tel que :

- ▷  $\forall i, j, \pi(i) \tilde{Q}_{ij} = \pi(j) \tilde{Q}_{ji}$
- ▷  $\forall i, \sum_{j \neq i} \tilde{Q}_{ij} = \sum_{j \neq i} Q_{ij}$

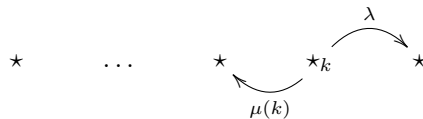
Alors  $(X_t)_t$  est ergodique, de probabilité invariante  $\pi$ , dont le processus de markov renversé est généré par  $\tilde{Q}$ .

**File M/M/∞**



$$\pi(k) = e^{-\frac{\lambda}{\mu}} \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!}$$

**“General” queue** Le débit de départ  $\mu(n)$  dépend de l'état  $n$  du système



$$\pi(k) = \pi(0) \frac{\lambda^k}{\prod_{i=1}^k \mu(i)}$$

### 6.5.1 Loss network

Soit  $G = (V, E)$  un graphe simple non-orienté. Soit  $R$  l'ensemble des routes possibles où une route est un chemin dans  $G$ . Les appels téléphonique arrivent suivant une loi de Poisson  $(\lambda_r)$ . La durée de l'appel de "type"  $r$  suit une loi exponentielle  $(\mu_r)$ .

**Version 1 :** Chaque lien (arrête a une capacité infinie). On prend pour espace d'état  $\{(n_r)_{r \in R} \mid n_r \in \mathbb{N}\}$ .

$$\pi((n_r)_{r \in R}) = \prod_{r \in R} e^{-\frac{\lambda_r}{\mu_r}} \left(\frac{\lambda_r}{\mu_r}\right) \frac{1}{n_r!}$$

**Version 2 :** Chaque lien a une capacité finie  $C_e \in \mathbb{N}^*, \forall e \in E$ . L'espace d'état  $\mathcal{E}$  est une restriction de l'espace d'état de la version précédente (cf. modèle erlang pour les files).

$$\pi((n_r)_{r \in R}) = \pi(0) \prod_{r \in R} \left(\frac{\lambda_r}{\mu_r}\right) \frac{1}{n_r!} \mathbb{1}_{(n_r)_{r \in R} \in \mathcal{E}}$$

$$\text{où } (n_r)_{r \in R} \in \mathcal{E} \iff \forall e \in E, \sum_{r|e \in R} n_r \leq C_e$$

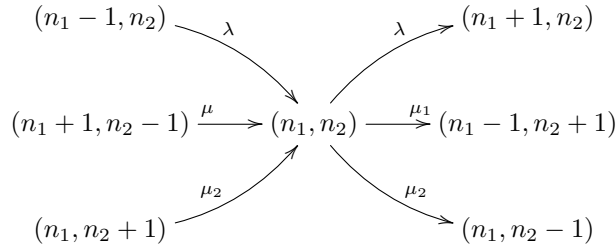
### 6.5.2 Tandem network

On considère la situation  $M/M/1/\infty \rightarrow \cdot/M/1/\infty$  dans laquelle les éléments sortant de la première queue rentrent dans la seconde. Les arrivées ont pour débit une loi de poisson de paramètre  $\lambda$ , les sorties de la première queue suivent une loi exponentielle de paramètre  $(\mu_1)$  et celle de la seconde queue une loi exponentielle de paramètre  $\mu_2$ .

$$\lambda < \min(\mu_1, \mu_2) \qquad \rho_1 = \frac{\lambda}{\mu_1} < 1 \qquad \rho_2 = \frac{\lambda}{\mu_2} < 1$$

On prend pour espace d'état  $\mathbb{N} \times \mathbb{N}$  où chaque composante indique le nombre d'élément dans leur queues respectives.

Le modèle est-il réversible? Non! Si on regarde les états  $(1, 0)$  et  $(0, 1)$ , on peut passer du premier au second mais pas réciproquement! On doit donc calculer à partir de l'équation d'équilibre global  $\nu Q = 0$



Pour chaque queue, on a pour probabilité invariante  $\pi(n_1, \star) = (1 - \rho_1)\rho_1^{n_1}$  et  $\pi(\star, n_2) = (1 - \rho_2)\rho_2^{n_2}$ , d'où en l'absence de corrélation, on aurait  $\pi(n_1, n_2) = \pi(n_1, \star)\pi(\star, n_2)$ . Posons  $\nu(n_1, n_2) = (1 - \rho_1)\rho_1^{n_1}(1 - \rho_2)\rho_2^{n_2}$  et calculons  $\nu Q$  :

$$\begin{aligned} \nu(n_1 - 1, n_2)\lambda + \nu(n_1 + 1, n_2 - 1)\mu_1 + \nu(n_1, n_2 + 1)\mu_2 &= \nu(n_1, n_2) [\lambda + \mu_1 + \mu_2] \\ (1 - \rho_1)\rho_1^{n_1}(1 - \rho_2)\rho_2^{n_2} \left[ \frac{\lambda}{\rho_1} + \frac{\rho_1}{\rho_2}\mu_1 + \rho_2\mu_2 \right] &= (1 - \rho_1)\rho_1^{n_1}(1 - \rho_2)\rho_2^{n_2} [\lambda + \mu_1 + \mu_2] \\ \frac{\lambda}{\rho_1} + \frac{\rho_1}{\rho_2}\mu_1 + \rho_2\mu_2 &= \lambda + \mu_1 + \mu_2 \\ \frac{\lambda}{\mu_1} + \frac{\frac{\lambda}{\mu_1}\mu_1}{\frac{\lambda}{\mu_2}} + \frac{\lambda}{\mu_2}\mu_2 &= \lambda + \mu_1 + \mu_2 \end{aligned}$$

**Théorème 6.5.** *Le modèle tandem est ergodique de probabilité invariante :*

$$\pi(n_1, n_2) = (1 - \rho_1)\rho_1^{n_1}(1 - \rho_2)\rho_2^{n_2}$$

En particulier, à tout temps  $t$  fixé, on a indépendance du nombre d'éléments dans les deux queues. Par contre entre deux instants donnés, on a pas indépendance : pour  $t_1$  et  $dt$  bien choisis ( $X_{t_1}^{(1)} = n_1, X_{t_1+dt}^{(1)} = n_1 - 1$ ) et ( $X_{t_2}^{(2)} = n_2, X_{t_2+dt}^{(2)} = n_2 + 1$ ).

**Théorème 6.6** (Burke (56)). *On considère une file  $M/M/1/\infty$ ,  $\rho = \frac{\lambda}{\mu} < 1$ ,  $X_0 \sim \pi$  ( $\pi(n) = (1 - \rho)\rho^n$ ). On pose  $D_n$  les instants de départ du  $n^{\text{ème}}$  client. Alors :*

- $(D_n)_n$  est un processus de poisson de débit  $\lambda$ .
- $X_t \Pi \{(D_n)_n, D_n < t\}$

**Démonstration :** Soit  $N$  les instants d'arrivées de  $(X_t)_t$ ,  $\tilde{N}$  les temps de départ de  $(X_t)_t$ ,  $\tilde{N}$  les instants d'arrivées de  $\tilde{X}$  et  $\tilde{N}'' = \tilde{N}$ . Grâce à la réversibilité,  $\tilde{X} \sim X$  et  $\tilde{N}$  est un processus de poisson de paramètre  $\lambda$ . Propriété des processus de poisson : Si  $\tilde{N}$  renversé dans le temps est encore un processus de Poisson de paramètre  $\lambda$  alors  $\tilde{N}$  est une loi de Poisson de paramètre  $\lambda$ . Pour le second énoncé,  $X_t$  est indépendant des temps d'arrivées après  $t$  donc en renversant le temps,  $X_t$  est indépendant des temps de départ avant  $t$ .

**Application aux tandem network :**  $X_t^{(2)}$  est une fonction de  $\{(A_n^{(2)}) \mid A_n^{(2)} < t\} = \{(D_n^{(1)}) \mid D_n^{(1)} < t\} \Pi X_t^{(1)}$  d'où  $X_t^{(1)} \Pi X_t^{(2)}$

### 6.5.3 Réseaux de Jackson

Le modèle est formé de  $K$  queues de type  $\cdot/M/1/\infty$  tel que les arrivées externes à la queue  $i$  est un processus de poisson de paramètre  $\hat{\lambda}_i \in [0, +\infty)$  et  $\exists i$  tel que  $\hat{\lambda}_i \neq 0$ . Les services suivent des lois exponentielles de paramètre  $\mu_i$  IID indépendantes entre elles et avec les arrivées. On peut rediriger un paquet sortant de la queue  $i$  vers l'entrée de la queue  $j$  avec probabilité  $r_{ij}$  et lui faire quitter le réseau avec la probabilité  $r_{i0} = 1 - \sum_j r_{ij}$ . On appelle  $R = (r_{ij})_{i,j \in \{1, \dots, K\}}$  la rooting matrix.

On prend deux hypothèse supplémentaires. D'une part, le réseau est "sans capture", c'est à dire qu'un client entrant dans le réseau peut en sortir, c'est à dire  $\rho(R) < 1$ . D'autre part,  $\forall i, r_{ii} = 0$  (ce qui se fait sans perte de généralité car si  $r_{ii} > 0$ , il suffit de changer  $\mu_i$  en  $r_{ii}\mu_i$  pour obtenir un résultat équivalent avec  $r_{ii} = 0$ ).

On pose  $R_{ij}^n$  la probabilité qu'un client entrant dans la queue  $i$  se retrouve dans la queue  $j$  au bout de  $n$  étapes. Si  $\mathcal{G}(R)$  est fortement connexe, alors  $R_{ij}^n \sim \rho(R)^n$  (Théorème de Penn-Frobenius).

**Tandem network :** Il y a deux queues avec  $\hat{\lambda}_1 \neq 0, \hat{\lambda}_2 = 0$ , et

$$R = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

**K queues consécutives**  $\hat{\lambda}_1 \neq 0, \hat{\lambda}_i = 0, \forall i > 1$ , et

$$R = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

**Réseaux de jackson :**  $X_t = (X_t^{(1)}, \dots, X_t^{(K)})$  généré par  $Q$ .  $\vec{n} = (n_1, \dots, n_K) \in \mathbb{N}^K$ ,  $e_i = (\delta_{ij})_{1 \leq j \leq K}$ .

$$\begin{cases} Q_{\vec{n}, \vec{n}+e_i} = \hat{\lambda}_i \\ Q_{\vec{n}, \vec{n}-e_i} = \mu_i r_{i0} & \text{si } n_i > 0 \\ Q_{\vec{n}, \vec{n}-e_i+e_j} = \mu_i r_{ij} & \text{si } n_i > 0 \text{ et } i \neq j \end{cases}$$

Si  $\exists i$  tel que  $\hat{\lambda}_i > \mu_i$ , le modèle est instable. Si  $\forall i, \hat{\lambda}_i < \mu_i$ , le système est-il stable? Pas forcément, il faut prendre en compte les apports internes.

**Définition 6.4** (Équation de Trafic).

$$\lambda_i = \hat{\lambda}_i + \sum_{j \neq i} \lambda_j R_{ji}$$

Intuitivement,  $\lambda_i$  correspond au débit total entrant dans la queue  $i$  et on s'attendrait à obtenir une condition de stabilité du genre  $\forall i, \lambda_i < \mu_i$ .

On passe en notations vectorielles :

$$\begin{aligned} \lambda &= \hat{\lambda} + \lambda R \iff \lambda [Id - R] = \hat{\lambda} \\ &\iff \lambda = \hat{\lambda} [Id - R]^{-1} = \hat{\lambda} \sum_{k \geq 0} R^k \end{aligned}$$

Pourquoi  $(Id - R)$  est-elle invertible?  $\lambda$  est une valeur propre de  $R \iff (1 - \lambda)$  est une valeur propre de  $(Id - R)$ . Or  $\rho(R) < 1$  donc toutes les valeurs propres de  $R$  sont strictement positives.

**Cas à 2 queues :**

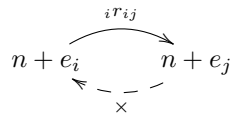
$$\lambda_1 = \frac{1}{1 - r_{12}r_{21}} [\hat{\lambda}_1 + r_{12}\hat{\lambda}_2] \qquad \lambda_2 = \frac{1}{1 - r_{12}r_{21}} [\hat{\lambda}_2 + r_{21}\hat{\lambda}_1]$$

**Théorème 6.7.** On considère un réseau de Jackson avec les notations précédentes.  $(X_t)_t$  est ergodique  $\iff \forall i, \lambda_i < \mu_i$  et alors la probabilité invariante est donnée par :

$$\pi((n_1, \dots, n_K)) = \prod_{i=1}^K (1 - \rho_i) \rho_i^{n_i} \qquad \rho_i = \frac{\lambda_i}{\mu_i}$$

- ▷ le processus de départ de la queue  $i$  vers l'extérieur suit une loi de Poisson de paramètre  $(\lambda_i r_{i0})$
- ▷ ces processus sont disjoints les uns des autres
- ▷  $X_t$  est disjoint des processus de départ vers l'extérieur jusqu'au temps  $t$ .
- ▷ On appelle un tel résultat un product form result
- ▷ À un temps  $t$  donné, il y a indépendance du nombre de clients dans chaque queues.
- ▷ Le processus des arrivées à la queue  $i$  n'est pas un processus de Poisson.
- ▷ Mais le processus des arrivées de l'extérieur à la queue  $i$  est un processus de Poisson de paramètre  $(\hat{\lambda}_i)$ .
- ▷ Le processus des départs à la queue  $i$  n'est pas un processus de Poisson.
- ▷ Mais le processus des départs de l'extérieur à la queue  $i$  est un processus de Poisson de paramètre  $(\lambda_i r_{i0})$ .

On a pas de réversibilité :  $r_{ij} > 0, r_{ji} = 0$



**Équation d'équilibre global** ( $\pi Q = 0$ )

$$\pi(n) \left[ \sum_i \hat{\lambda}_i + \sum_{i|n_i>0} \mu_i \right] = \sum_{i|n_i>0} \pi(n - e_i) \hat{\lambda}_i + \sum_i \sum_{j|n_j>0} \pi(n + e_i - e_j) \mu_i r_{ij} + \sum_i \pi(n + e_i) \mu_i r_{i0}$$

Cette équation est difficile à résoudre, on commence par introduire des équations d'équilibre locales.

**Au niveau de la queue  $i$  ( $n_i > 0$ ) :**

$$\pi(n) \mu_i \stackrel{?}{=} \pi(n - e_i) \hat{\lambda}_i + \sum_{j \neq i} \pi(n + e_j - e_i) \mu_j r_{ji}$$

**Vers l'extérieur :**

$$\pi(n) \left[ \sum_i \hat{\lambda}_i \right] \stackrel{?}{=} \sum_i \pi(n + e_i) \mu_i r_{i0}$$

Si on a une solution à ces équations, on obtient une solution à l'équation globale en sommant les équations.

On fait une seconde hypothèse "magique" qui est de supposer  $\pi$  de la forme d'un produit comme dans le théorème. On substitue dans les équations "locales" :

$$\left( \prod_k (1 - \rho_k) \rho_k^{n_k} \right) \mu_i \stackrel{?}{=} (1 - \rho_i) \rho_i^{n_i - 1} \prod_{k \neq i} (1 - \rho_k) \rho_k^{n_k} \hat{\lambda}_i + \sum_{j \neq i} \left( (1 - \rho_j) \rho_j^{n_j - 1} (1 - \rho_i) \rho_i^{n_i - 1} \prod_{\substack{k \neq i \\ k \neq j}} (1 - \rho_k) \rho_k^{n_k} \right) \mu_j r_{ji}$$

$$\mu_i \stackrel{?}{=} \frac{1}{\rho_i} \hat{\lambda}_i + \dots$$

$$\lambda_i \stackrel{?}{=} \hat{\lambda}_i + \sum_{j \neq i} \lambda_j r_{ji}$$

Or la dernière équation est l'équation de trafic qui est vérifiée par définition.  $\pi$  est donc bien une solution à l'équation d'équilibre globale.

Pour la seconde partie du théorème, on considère le processus renversé de  $(X_t)_t$  :

$$\tilde{\mathbb{Q}} \rightarrow (\tilde{X}_t)_t \rightsquigarrow (X_t)_t \leftarrow \mathbb{Q}$$

$$\tilde{\mathbb{Q}}_{n, n+e_i} = \frac{\pi(n + e_i)}{\pi(n)} \mathbb{Q}_{n+e_i, n} = \frac{\lambda_i}{\mu_i} \mu_i r_{i0} = \lambda_i r_{i0}$$

$$\tilde{\mathbb{Q}}_{n+e_i, n+e_j} = \frac{\pi(n + e_j)}{\pi(n + e_i)} \mathbb{Q}_{n+e_j, n+e_i} = \frac{\lambda_j}{\mu_j} \frac{\mu_i}{\lambda_i} r_{ji} = \frac{\lambda_j}{\lambda_i} r_{ji}$$

$$\tilde{\mathbb{Q}}_{n+e_i, n} = \frac{\pi(n)}{\pi(n + e_i)} \mathbb{Q}_{n, n+e_i} = \frac{\mu_i}{\lambda_i} \hat{\lambda}_i$$

$\tilde{\mathbb{Q}}$  est le générateur infinitésimal d'un autre réseau de Jackson que l'on appelle le *réseau de Jackson renversé*. En effet, on peut introduire la rooting matrix  $\tilde{R}$  :

$$\tilde{R}_{ij} = \frac{\lambda_j}{\lambda_i} r_{ji} \qquad \tilde{r}_{i0} = \frac{\hat{\lambda}_i}{\lambda_i}$$

et l'équation suivante est vérifiée car elle correspond à l'équation de trafic :

$$\sum_{j \neq i} \tilde{R}_{ij} + r_{i0} \stackrel{?}{=} 1$$

$$\sum_{j \neq i} \frac{\lambda_j}{\lambda_i} r_{ji} + \frac{\hat{\lambda}_i}{\lambda_i} = 1$$

## 6.6 Réseau de Gordon-Newell

(Réseau de Jackson clos)

On se place dans un cadre où l'on a  $K$  queues,  $M$  clients, la matrice de redirection (routing matrix)  $R$  est stochastique ( $\rho(R) = 1$ ) et inversible. On utilise les mêmes notations que précédemment.  $X$  est le nombre de client dans les queues  $X = (n_1, \dots, n_K)$ . L'espace d'état est  $\mathcal{E} = \{(n_1, \dots, n_K) \mid \sum_i n_i = M\}$ .  $Q$  est le générateur infinitésimal défini par  $Q_{n+e_i, n+e_j} = \mu_i r_{ij}$ .

**Équation de trafic :**  $\nu_i$  est le flux à travers la station  $i$ .

$$\forall i, \nu_i = \sum_{j \neq i} \nu_j R_{ji}$$

Sous forme vectorielle, on obtient  $\nu = \nu R$  avec  $R$  stochastique donc on peut l'interpréter comme la matrice de transition d'une chaîne de Markov discrète pour laquelle  $\nu$  est une distribution stationnaire donc unique à une constante multiplicative près.

**Théorème 6.8.**  $(X_t)_t$  est toujours ergodique et  $\forall n = (n_1, \dots, n_K) \in \mathcal{E}$

$$\pi(n) = G^{-1} \prod_{k=1}^K \left( \frac{\nu_k^{n_k}}{\mu_k} \right)$$

$$G = \sum_{n \in \mathcal{E}} \prod_{k=1}^K \mu_k^{n_k}$$

## Troisième partie

# Game theory

### 6.7 $M/M/1/\infty$ :

Soit  $\lambda < \mu$  et  $\pi(n) = (1 - \rho)\rho^n$  avec  $\rho = \frac{\lambda}{\mu}$ .  $W$  est le temps passé par les clients dans la queue.

$$W = \underbrace{S}_{Exp(\mu)} + \sum_{j=1}^L \underbrace{S_j}_{Exp(\mu)}$$

avec  $L \sim \pi$  par la propriété PASTA (Poisson Arrivals See Time Average).

**Proposition 6.6.**  $W = Exp(\mu - \lambda)$

**Équilibre de Nash :** On dit qu'un système est à l'équilibre de Nash si aucun client n'a intérêt à changer de route dans le réseau.

**Paradoxe de Brass :** L'équilibre optimal pour les utilisateurs est distinct de l'équilibre optimal du système. En terme de théorie des jeux, "L'équilibre de Nash n'est pas Pareto optimal".

## 7 Finite games (atomic games)

Chaque joueur est un atome et son action sur le jeu est visible.

**Définition 7.1.**  $G = (\mathcal{N}, S, U)$  où  $\mathcal{N}$  est le nombre de joueur (fini),  $S$  le nombre de stratégies (fini) et  $U = (U_i(s_1, \dots, s_n))_{i=1 \dots \mathcal{N}}$  est le  $\mathcal{N}$ -upplet des utilités de chaque joueur lorsque ceux-ci jouent les stratégies  $s_1, \dots, s_n$

**Notation :**  $U(s_1, \dots, s_n) = U_i(s_i, s_{-i})$

**Définition 7.2.** Un équilibre de Nash est une stratégie  $s$  tel que  $\forall i U_i(s) \geq U_i(s'_i, s'_{-i}), \forall s'_i \in S$

**The mixed extension of games :** Soit  $\sigma_i$  une mesure sure  $s_i$

$$U_i(\sigma_1, \dots, \sigma_{\mathcal{N}}) \stackrel{\text{def}}{=} \sum_s U_i(s) \sigma_1(s_1) \dots \sigma_{\mathcal{N}}(s_{\mathcal{N}})$$

**Théorème 7.1** (théorème principal). *Tout jeu a un équilibre de Nash dans son extension mixte.*

**Théorème 7.2** (théorème du point fixe de Kabutani). *Soit  $S$  un compact fermé de  $\mathbb{R}^n$  et  $\Phi : S \rightarrow 2^S$  tel que le graphe de  $\Phi(s)$  est fermé et convexe  $\forall s$  alors  $\exists x \in S$  tel que  $x \in \Phi(x)$ .*

Pour montrer l'existence d'un équilibre de Nash, on applique le théorème de Kabutani à la fonction "best response" :

$$BR_i : \begin{cases} S & \longrightarrow & 2^{S_i} \\ s & \longmapsto & \arg \max_{x_i} U_i(x_i, s_{-i}) \end{cases}$$

$$b_i(\sigma) = \{y \in \mathcal{P}(S_i) \mid y_k = 0 \text{ si } k \notin BR_i(\sigma)\}$$

$$b(\sigma) = \prod_i b_i(\sigma)$$

### Polynomial Parity Arguments in directed graphs : (PPAD)

- ▷ Soit un graphe orienté tel que les nœuds ait au plus un arc entrant et un arc sortant.
- ▷ Il existe un nœud  $s$  sans arc entrant.
- ▷ On peut calculer le voisinage d'un sommet en tant polynomial.

Trouver un nœud sans arc entrant ni arc sortant distinct de  $s$ .

**Théorème 7.3.** *Calculer l'équilibre de Nash d'un jeu  $G$  est PPAD-complete.*

**Best response algorithm :** Pour calculer l'équilibre de Nash

Commencer au point  $s$

**repeat**

Choisir un joueur ( $i$ )

$i$  choisit  $s_i \in BR_i(s)$

**until** bloqué

Si le jeu est un jeu à potentiel, l'algorithme "best response" calcule un équilibre de Nash en temps fini.

On dit qu'un jeu est à potentiel s'il existe  $F$  tel que

$$\forall i, U_i(s_i, s_{-i}) - U_i(s'_i, s'_{-i}) = F(s_i, s_{-i}) - F(s'_i, s'_{-i})$$

*Exercice 7.0.1.* Trouver un jeu à potentiel qui contient un équilibre de Nash que l'algorithme "best response" ne trouve pas.

## 8 Non-atomic games

Il y a beaucoup de joueurs qui forment une population. On suppose pour l'instant qu'il y a qu'une seule population (cette hypothèse sera supprimée plus tard). La *masse* de la population est 1. Les joueurs ont un nombre fini de stratégies  $(1, \dots, S)$  et on suppose que chaque joueur joue une unique stratégie. On note  $x_k$  la masse des joueurs jouant la stratégie  $k$  ( $\sum_{k=1}^S x_k = 1$ ). Le gain d'un joueur choisissant la stratégie  $i$  est  $F_i(x_1, \dots, x_S)$ .

$$BR(x) = \{i \mid F_i(x) \geq F_j(x) \forall j\}$$

$$b(x) = \left\{ y \mid \sum_i y_i = 1, y_j = 0 \text{ si } j \notin BR(x) \right\}$$

**Définition 8.1.**  $x \in NE(F) \iff x \in b(x)$  (NE = Nash equilibrium)

**Théorème 8.1.** *Il existe toujours un équilibre de Nash (Wardrop equilibrium).*

Si  $x \in NE$  alors  $F_i(x) = F_j(x)$  si  $x_i > 0$  et  $x_j > 0$

### 8.1 Jeu à populations multiples

$p \in P$  est une population de masse  $m_p$ . On adapte les notations pour le gain :  $F_i^p(x^p, x^{-p}) =$  utilité d'un joueur appartenant à la population  $p$  qui joue la stratégie  $i$ .  $x^p = (x_1^p, \dots, x_{S_p}^p)$  avec  $\sum x_i^p = m_p$ .

#### 8.1.1 Jeux de congestion

On considère un réseau avec des paires de sommets source/but. On note  $e$  les arrêtes, une route dans le réseau est un ensemble  $E_i^k$  d'arrêtes utilisées par le flot  $k$  (= la population  $k$  correspondant à la  $k$ -ème paire source/but). La proportion des paquets du flot  $k$  passant par la route  $i$  est  $x_i^k$  et  $x_e = \sum_k \sum_i$  tel que  $e \in E_i^k$   $x_i^k$  est la masse des paquets utilisant l'arrête  $e$ . L'utilité est alors donnée par  $F_i^k(x) = - \sum_{e \in E_i^k} l_e(x_e)$  où  $l_e$  est le délai sur l'arrête  $e$ .

#### 8.1.2 Jeux à potentiel

**Définition 8.2.**  $\exists f$  tel que  $\frac{\partial f}{\partial x_i^p}(x) = F_i^p(x) \iff \frac{\partial F_i^p}{\partial x_j^q} = \frac{\partial F_j^q}{\partial x_i^p}$

**Exemple :** Dans le cas du jeu de congestion considéré précédemment, on a :

$$\frac{\partial F_i^p}{\partial x_j^q} = \sum_{e \in E_i^p \cap E_j^q} l'_e(x_e) = \frac{\partial F_j^q}{\partial x_i^p}$$

Il s'agit donc bien d'un jeu à potentiel, avec pour potentiel :

$$f(x) = \sum_e \int_0^{x_e} l_e(z) dz$$

**Définition 8.3.** Le *prix de l'anarchie* est le rapport

$$\frac{\sup \left( \sum_p \sum_i x_i^p U_i(x_i) \right)}{\inf_{x^w} \sum_p \sum_i x_i^{pw} U_i^p(x^w)} > 1$$



Quatrième partie

## Price of anarchy in multiserver queue