

# Prendre les bonnes décisions avec les Processus markoviens décisionnnels

**Tiffany Cherchi**

*Doctiss 2019, Montpellier*

**THALES**



**Inria**  
informatiques mathématiques

# Le quotidien des doctorant.e.s

Sleep

Biblio

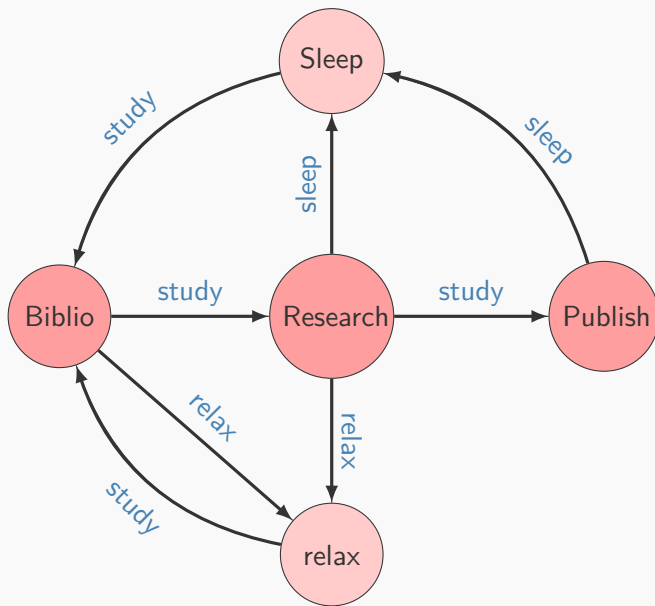
Research

Publish

États

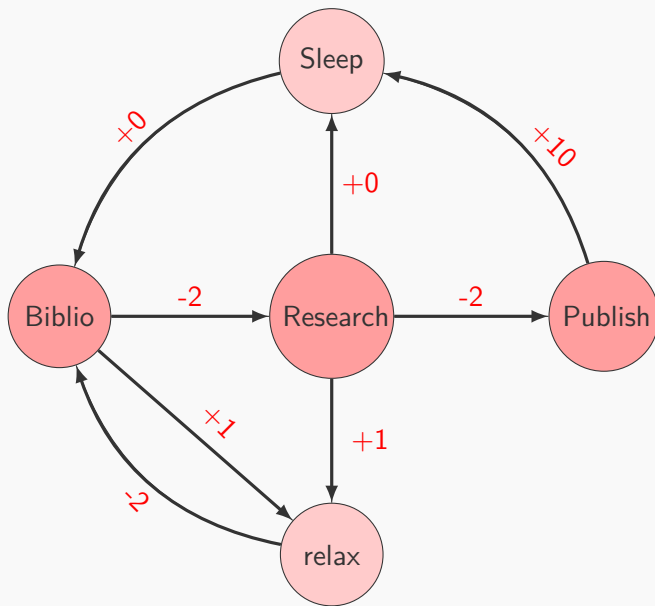
relax

# Le quotidien des doctorant.e.s



États  
**Actions**

# Le quotidien des doctorant.e.s



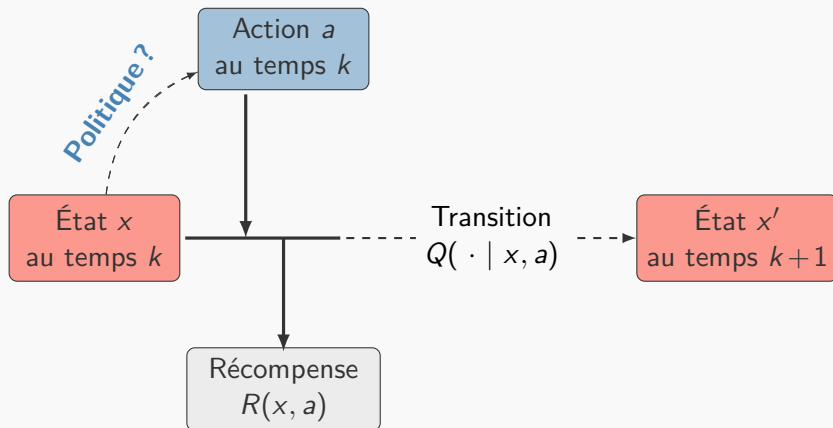
États  
Actions  
**Récompenses**

# Définition d'un MDP

Un processus markovien décisionnel est défini par :

- ▶ L'ensemble des états,  
 $\mathbb{X} = \{biblio, research, publish, relax, sleep\}.$
- ▶ L'ensemble des actions,  
 $\mathbb{A} = \{relax, study, sleep\}.$
- ▶ La fonction de récompenses  $R$  à valeurs réelles dépendant du couple (état  $x$ , action  $a$ ) .
- ▶ Un noyau de transition  $Q(\cdot \mid x, a)$  qui permet de calculer les transitions aléatoires du système lorsque celui-ci se trouve dans l'état  $x$  et que l'action  $a$  est choisie.

# Construction du processus



- Comment choisir ma suite d'actions ?
- Quelle meilleure *récompense cumulée* puis-je espérer obtenir ?

# Qu'est-ce qu'une politique ?

	Biblio	Research	Publish	Relax	Sleep
<b>Step 1</b>	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
<b>Step 2</b>	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
<b>Step 4</b>	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$
...	...	...	...	...	...
<b>Step N</b>	$a_{N,1}$	$a_{N,2}$	$a_{N,3}$	$a_{N,4}$	$a_{N,5}$

**Table – Politique à horizon fini  $N$ .**

# Problème de contrôle optimal

La **récompense totale** partant de l'état  $x$  et suivant la politique  $\pi$  jusqu'à l'horizon  $N$  est

$$v_N(\pi, x) = \mathbb{E} \left[ \sum_{n=0}^N R(x_n, a_n) \mid \pi, x \right].$$

Le problème est de **maximiser**, sur l'ensemble des politiques admissibles  $\Pi$ , la fonction :

$$\pi \rightarrow v_N(\pi, x).$$

L'**optimum** est donné par :  $v(x) = \sup_{\pi \in \Pi} v_N(\pi; x).$

La politique  $\pi^* \in \Pi$  est **optimale** si elle vérifie  $V_N(\pi^*, x) = V(x).$



# Programmation dynamique

---

## Algorithme 1 : Programmation dynamique

---

Entrées : États  $\mathbb{X}$ , actions  $\mathbb{A}$ , transitions  $Q$ , Récompenses  $R$

```

1  début
2    pour tout  $x \in \mathbb{X}$  faire
3       $v[N + 1, x] = 0$ 
4    pour  $k$  de  $N$  à 0 faire
5      pour tout  $x \in \mathbb{X}$  faire
6         $v[k, x] = \max_{a \in \mathbb{A}(x)} \left[ R(x, a) + \sum_{y \in \mathbb{X}} v[k + 1, y] Q(y \mid x, a) \right]$ 
7         $\pi[k, x] = \operatorname{argmax}_{a \in \mathbb{A}(x)} \left[ R(x, a) + \sum_{y \in \mathbb{X}} v[k + 1, y] Q(y \mid x, a) \right]$ 
8    retourner  $v_0, \pi^*$ 
  
```

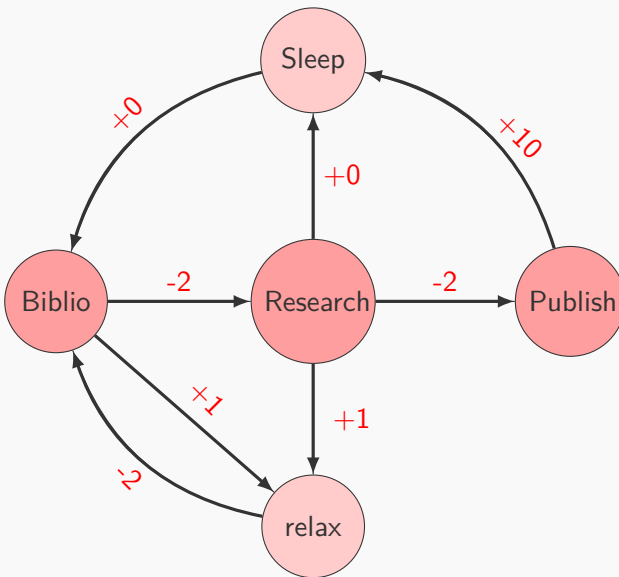
---

# Résultats numériques

	Biblio	Research	Publish	Relax	Sleep
<b>R max</b>	<b>6</b>	<b>8</b>	<b>11</b>	<b>-3</b>	<b>-1</b>
<b>Policy</b>					
<b>Step 1</b>	Study	Study	Sleep	Study	Study
<b>Step 2</b>	Relax	Study	Sleep	Study	Study
<b>Step 3</b>	Relax	Relax	Sleep	Study	Study

**Table – Récompense et politique optimales, pour un horizon  $N=3$**

## Vérifions :



# Résultats numériques

	Biblio	Research	Publish	Relax	Sleep
<b>R max</b>	<b>7</b>	<b>7</b>	<b>16</b>	<b>4</b>	<b>6</b>
<b>Policy</b>					
<b>Step 1</b>	Study	Study	Sleep	Study	Study
<b>Step 2</b>	Study	Study	Sleep	Study	Study
<b>Step 3</b>	Study	Study	Sleep	Study	Study
<b>Step 4</b>	Relax	Study	Sleep	Study	Study
<b>Step 5</b>	Relax	Relax	Sleep	Study	Study

**Table** – *Récompense et politique optimales, pour un horizon  $N=5$*

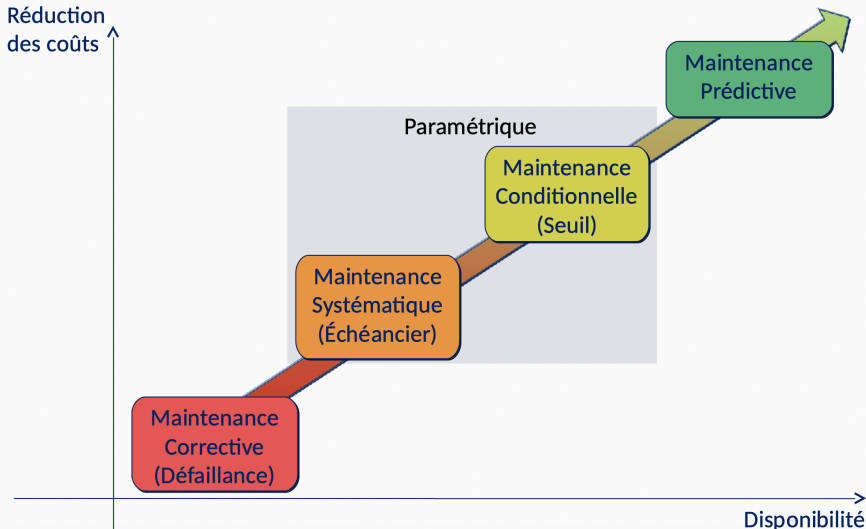
## Conclusions

- ▶ Outil pour modéliser la prise de décision séquentielle.
- ▶ Méthode numérique pour résoudre le problème d'optimisation.

## Dans le cadre de ma thèse

- ▶ Thèse Cifre avec l'entreprise Thales.
- ▶ Application à de l'optimisation de maintenance.

# Évolution de la Maintenance



# Problématique industrielle

## Modéliser la dynamique d'équipements

- ▶ à plusieurs composants, et plusieurs états,
- ▶ requis pour des missions, sujets à des pannes aléatoires,
- ▶  $\mathbb{X} = \{x = (e_i, r_i); e_i \in \{\text{stable, dégradé, panne}\}, r_i \in \mathbb{R}^+\}$ .

## Trouver une politique de maintenances ..

- ▶ quelle action : mission / atelier ( entretenir ou remplacer) ?
- ▶ quand ?

## .. qui optimise un certain critère

- ▶ **minimiser** coûts de maintenance
  - ▶ **maximiser** la disponibilité
- } **Compromis non trivial**

**Problèmes :** Espace d'états infini, et noyau non explicite :

~~tc[Red]~~ PROGRAMMATION DYNAMIQUE

# Merci de votre attention !

Tiffany Cherchi  
*Doctiss 2019, Montpellier*

THALES



informatiques mathématiques  
*Inria*