Processus décisionnels markoviens

1 Le problème du parking

On souhaite se garer au plus près d'un cinéma, afin de minimiser le temps de marche nécessaire pour y entrer (voir figure 1). Le parking attenant au cinéma est constitué d'une suite de n rangées parallèles : arrivé devant l'une d'entre elle, on peut décider soit d'essayer de s'y garer, soit de continuer à s'avancer. Si l'on essaye de se garer dans une rangée, on a 30 % de chance d'y trouver une place libre. En cas d'échec (70 % de chance), on est contraint de se rendre devant la rangée suivante. Si l'on n'a pas réussi à se garer dans le parking attenant, on doit obligatoirement se garer dans le parking extérieur, où il y a toujours de la place, mais qui ne se situe pas directement à côté. Le problème est de savoir, lorsque l'on arrive devant une rangée, s'il vaut mieux essayer de se garer, ou continuer à s'avancer.

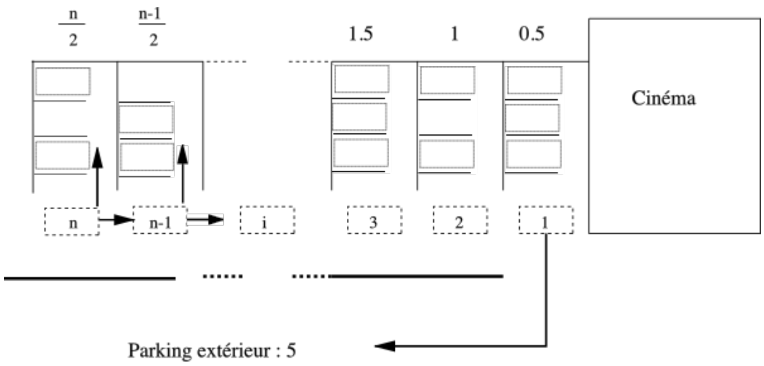

FIGURE 1 – Représentation schématique du problème du parking. Les valeurs indiquées au dessus des rangées et pour le parking extérieur correspondent aux temps de marche pour rejoindre le cinéma (en minutes).

Question 1 : Définir les états dans lesquels l'agent peut se trouver, les actions possibles dans chaque état, les probabilités de transition entre les états et les coûts induits par les actions. Pour cette formalisation, on se ramènera au cas d'un problème à **horizon infini** (considérer que lorsque l'agent atteint un état terminal, il reste dans cet état avec une probabilité de 1 et un coût de transition nul).

Question 2 : Représenter le problème sous forme de graphe.

Question 3 : Appliquer l'algorithme d'itérations sur les valeurs. Prendre $\gamma = 1$ (pour simplifier les calculs) et $n = 5$. On arrêtera l'itération quand $V_{t+1}(i) = V_t(i)$ pour tout état i (c.-à-d. quand $\|V_{t+1} - V_t\| = 0$).

Question 4 : Quelle est la politique optimale ?

2 Le problème du chauffeur de taxi

Un chauffeur de taxi travaille dans trois quartiers A, B et C. Ses courses peuvent l'amener à faire un trajet au sein d'un quartier, ou entre deux quartiers. Quand il doit trouver un nouveau client pour sa prochaine course, trois options s'offrent à lui. Il peut :

— a_1 : rouler dans le quartier actuel en espérant être hélé par un client,

— a_2 : s'arrêter à une station de taxis dans le quartier actuel et attendre un client,

— a_3 : stationner son taxi dans le quartier actuel et attendre un appel.

Dans les quartiers A et C, les trois actions sont possibles ; en revanche dans le quartier B, seules a_1 et a_3 sont possibles (il n'y a pas de station de taxis dans B). Les probabilités de transition et les récompenses en euros (p.ex. les bénéfices moyens attendus en fonction du quartier et de la méthode) sont fournies par le tableau 1.

Table 1 – Probabilités de transition (à gauche), et récompenses (à droite).

P		action	quartier suivant A	B	C
quartier actuel	A	a_1	1/2	1/4	1/4
		a_2	1/16	3/4	3/16
		a_3	1/4	1/8	5/8
	B	a_1	1/2	0	1/2
		a_3	1/16	7/8	1/16
	C	a_1	1/4	1/4	1/2
		a_2	1/8	3/4	1/8
		a_3	3/4	1/16	3/16

		action	R
quartier actuel	A	a_1	22
		a_2	14
		a_3	14
	B	a_1	32
		a_3	32
	C	a_1	20
		a_2	12
		a_3	12

Question 1 : Quels sont les différents états du problème ?

Question 2 : Produire le graphe des transitions partant de B uniquement.

Question 3 : Appliquer l'algorithme d'itération sur les politiques jusqu'à l'écriture du premier système d'équations linéaires. Prendre $\gamma = 0{,}8$. Pour l'initialisation, on utilisera la politique arbitraire suivante : $\forall s \in \mathcal{S}, \mu_0(s) = a_1$.

Question 4 : Ce système peut-être résolu de façon automatique, par exemple en Python avec la bibliothèque `numpy` :

```
>>> a = np.array([[-3/5, 1/5, 1/5], [2/5, -1, 2/5], [1/5, 1/5, -3/5]])
>>> b = np.array([-22, -32, -20])
>>> np.linalg.solve(a, b)
array([115.41666667, 123.33333333, 112.91666667])
```

Comment interpréter le résultat ?