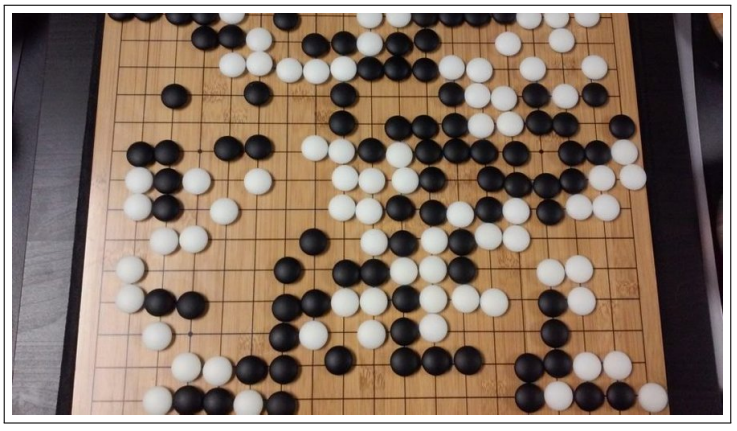
Faculté des Sciences & Ecole des Mines de Nancy

Apprentissage par renforcement

Génie Industriel & Mathématiques Appliquées (GIMA)

Année universitaire 2024-2025 - Semestre S9

Cours dispensé par Pascal MOYAL & Yannick PRIVAT



Dép. Génie Industriel et Mathématiques Appliquées
Ecole des Mines de Nancy - Campus Artem
54 042 Nancy Cedex
emails : yannick.privat@univ-lorraine.fr

Table des matières

Préambule

Voici le polycopié du cours d'Apprentissage par renforcement. Il s'agit d'un cours introductif composé de deux parties :

- ⇝ la programmation dynamique, à la base de nombreux algorithmes ;
- ⇝ l'apprentissage par renforcement.

Ce cours doit être considéré comme une base qui vous permettra, si vous le souhaitez ou que vous en avez besoin dans le cadre d'un projet, de compléter vos connaissances sur le sujet. Un ensemble de concepts, principalement dérivés des programmes de L3, sont rappelés au sein d'appendices.

Le cours sera évalué de la façon suivante :

- Un examen écrit (de 2 heures) le 4 février 2025 ;
- Un exposé à réaliser en binôme le 11 février 2025, en effectuant des recherches. La présentation durera 15 min (hors questions) et vous remettrez un cours rapport résumant les pistes étudiées.

  Vous coderez un des problèmes proposés ci-après et réaliserez une présentation dans laquelle :
  - vous expliciterez très soigneusement la modélisation du problème d'optimisation sous-jacent (espace d'état, ensemble des actions, dynamique éventuelle du système / matrices de transition, récompense, politique, etc.)
  - vous présenterez l'algorithme choisi et ses étapes principales ; vous avez le droit d'utiliser des bibliothèques existantes ;
  - vous exécuterez cet algorithme et discuterez des résultats.
  - vous mentionnerez toutes vos sources (incluant les sites web et les codes recopiés. Il n'est pas interdit d'utiliser un code déjà écrit si vous le signalez et que vous êtes en mesure de l'expliquer en détails).

1. **Apprentissage d'un agent sur CartPole** : Entraînez un agent à équilibrer un bâton sur un chariot avec Q-learning ou Deep Q-Networks (DQN) à l'aide de la bibliothèque `OpenAI Gym`.
2. **Robot de navigation simple** : Implémentez un agent pour trouver la sortie dans un labyrinthe 5x5. Utilisez une table Q et le Q-learning pour optimiser la navigation.
3. **Problème des bandits multi-bras** : Simulez un problème avec 5 machines à sous (bandits) et résolvez-le avec UCB (Upper Confidence Bound). Visualisez les gains cumulés.
4. **Balayage avec un aspirateur virtuel** : Entraînez un agent à nettoyer une pièce en minimisant les mouvements. Implémentez SARSA dans un environnement à grille.
5. **Optimisation d'une file d'attente** : Créez une simulation où un agent gère une file d'attente en minimisant le temps d'attente moyen. Testez avec Q-learning.
6. **Gestion de stock basique** : Modélisez un problème simple de gestion d'inventaire où l'agent décide combien commander pour minimiser les coûts. Implémentez SARSA.
7. **Jeu du Tic-Tac-Toe** : Entraînez un agent à jouer au Tic-Tac-Toe contre un joueur humain ou une IA simple. Utilisez une politique ε-greedy avec une table Q.
8. **Navigation avec un vent aléatoire** : Simulez un environnement 5x5 où un agent doit atteindre une destination malgré des déplacements perturbés. Implémentez Q-learning.
9. **Contrôle des feux de circulation** : Simulez un carrefour avec deux feux. L'agent doit réduire le temps d'attente total en utilisant Q-learning.

10. **Course vers une cible mobile** : Implémentez un agent qui poursuit une cible se déplaçant aléatoirement dans une grille. Utilisez une table Q pour optimiser sa trajectoire.
11. **Gestion d'une station de recharge** : Modélisez une station où un agent décide de charger ou d'attendre pour minimiser les coûts énergétiques. Implémentez Monte Carlo.
12. **Jeu du Snake simplifié** : Implémentez une version réduite du jeu Snake où l'agent apprend à éviter les murs et à manger des pommes. Utilisez DQN.
13. **Problème du taxi** : Entraînez un agent à ramasser et déposer des passagers sur une grille en minimisant le temps de trajet. Utilisez par exemple un algorithme de Q-learning.
14. **Agent de tri de déchets** : Implémentez un agent qui trie des déchets (plastique, verre, papier) arrivant en séquence. Utilisez une politique simple avec ε-greedy.
15. **Clic sur des annonces publicitaires** : Simulez un environnement où l'agent choisit entre plusieurs annonces pour maximiser les clics. Implémentez une méthode de bandits multi-bras.

Nancy, le 1er décembre 2024, Pascal Moyal & Yannick Privat

I.

Introduction

I.1 Généralités et exemples

L'*apprentissage par renforcement* est un domaine de recherche qui se concentre sur le développement d'algorithmes capables d'apprendre automatiquement des stratégies, appelées "politiques". Elles déterminent les actions appropriées à entreprendre par un agent en fonction de ses observations de l'environnement pour optimiser un critère appelé "récompense". Ainsi, on parle :

- d'"apprentissage" parce que l'agent améliore progressivement sa stratégie (politique) en interagissant avec l'environnement. À partir de ses expériences, il ajuste ses décisions pour maximiser les récompenses cumulées, sans connaissance préalable des règles ou dynamiques du système. Ce processus adaptatif repose sur l'exploration, l'observation et la mise à jour de ses estimations.
- de "renforcement" parce que l'agent apprend en renforçant les actions qui mènent à des récompenses positives et en évitant celles qui génèrent des punitions. Ce principe s'inspire du conditionnement opérant en psychologie, où les comportements sont façonnés par leurs conséquences. L'objectif est de maximiser les récompenses cumulées sur le long terme en ajustant progressivement la stratégie de décision. À chaque interaction avec l'environnement, l'expérience de l'agent renforce ou affaiblit ses choix.

Ce domaine se distingue de celui de la *commande optimale* (ou *contrôle optimal*), qui traite également de la résolution de problèmes similaires, mais dans un cadre où l'on dispose généralement de davantage d'informations sur l'environnement (en général, des équations sur le système physique, biologique ou économique considéré). En général, l'apprentissage par renforcement propose des outils très génériques, mais qui offrent souvent peu de garanties, tandis que la commande optimale a élaboré des algorithmes plus spécifiques, dont la convergence est connue.

Ces deux domaines partagent un fondement commun : **la programmation dynamique**, qui fournit un cadre général pour résoudre des problèmes de prise de décision séquentielle, après observation de l'état d'un système. Cette approche peut être appliquée aussi bien à des systèmes classiques de contrôle, comme le pilotage d'un bras robotisé en choisissant la tension appliquée aux moteurs en fonction de la position observée, qu'à des problèmes dans des contextes probabilistes, tels que la finance, où il s'agirait de décider quand acheter ou vendre une action en fonction de l'évolution de son prix. Elle s'applique également à des problématiques en intelligence artificielle, comme le choix du coup à jouer dans une partie d'échecs en fonction de la position des pièces sur l'échiquier.

Voici un exemple de problème de contrôle optimal.

Rendez-vous de deux vaisseaux spatiaux au voisinage de la terre. Supposons que le véhicule 1 est passif et de trajectoire circulaire, et le véhicule 2 est activé par un moteur exerçant une poussée $v = (v_1, v_2)$ pour rattraper le véhicule 1. Désignons par x le vecteur linéarisé de la position du véhicule 2 dans le repère mobile d'origine le véhicule 1. Alors $x = (x_1, x_2)$ obéit aux équations de Hill

$$\begin{cases} \ddot{x}_1(t) = 3\omega^2 x_1(t) + 2\omega \dot{x}_2(t) + v_1(t) \\ \ddot{x}_2(t) = -2\omega \dot{x}_1(t) + v_2(t) \\ (x_1(0), \dot{x}_1(0), x_2(0), \dot{x}_2(0)) = X_0 \in \mathbb{R}^4 \end{cases}$$

Posons $X_v = (x_1, \dot{x}_1, x_2, \dot{x}_2)$, où les x_i sont les solutions du système ci-dessus, autrement dit les positions et vitesses relatives des vaisseaux dans le repère mobile. La pulsation ω correspond à la période de révolution

$T = 2\pi/\omega$ du véhicule 1, elle est égale à 5480 secondes pour la station spatiale internationale. Modélisons le problème. Pour effectuer un rendez-vous spatial, plusieurs approches sont possibles :

(i) *Temps minimum.* On prend en compte la puissance du moteur en introduisant l'ensemble (admissible) de contraintes sur le contrôle

$$\mathscr{U}_{\text{ad}} = \{v \in C^0([0,T]) \mid v(s) \in U\},$$

où U est un compact de $\mathbb{R}^2$, par exemple, on suppose que les moteurs sont indépendants tels que $|v_i(s)| \leq 1$, $i = 1,2$. On cherche un contrôle v qui ramène le vaisseau de X_0 à $X(T) = 0$ en un temps minimal T, autrement dit

$$\textit{minimiser } T \textit{ tel que } X_v(T) = 0 \textit{ parmi les contrôles } v \in \mathscr{U}_{\text{ad}}.$$

(ii) *Compromis énergie-précision.* Cette fois, on fixe un horizon de temps $T > 0$, et on souhaite que les deux vaisseaux se rencontrent à l'horizon de temps T. Une façon de procéder consiste à définir la fonctionnelle

$$J(v) = \frac{\varepsilon}{2}\int_0^T \left(v_1(t)^2 + v_2(t)^2\right) dt + (1-\varepsilon)(x_1(T)^2 + \dot{x}_1(T)^2 + x_2(T)^2 + \dot{x}_2(T)^2)$$

où $\varepsilon \in]0,1[$ est un poids donné. Le premier terme de la somme traduit le "coût du contrôle" tandis que le second cherche à rendre les positions et vitesses finales relatives nulles.

On cherche à minimiser J sur un certain ensemble des contrôles admissibles $\mathscr{U}_{\text{ad}}$, i.e. on résout

$$\textit{minimiser } J(v) \textit{ parmi les contrôles } v \in \mathscr{U}_{\text{ad}}.$$

On remarque que dans cette approche, la cible finale $X_v(T)$ n'est pas forcément atteinte, mais se présente dans la fonction coût comme une pénalisation.

Un problème d'apprentissage par renforcement. Considérons où un robot mobile doit se déplacer dans une grille 5×5 pour atteindre une destination tout en évitant des obstacles. Le robot peut se déplacer dans quatre directions : haut, bas, gauche, droite. À chaque étape, il reçoit une récompense en fonction de son action et de sa position.

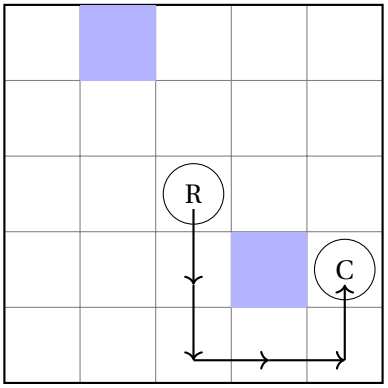


FIGURE I.1 : *Robot R, Cible C et un exemple de commande permettant d'atteindre la cible. Les parois colorées sont des obstacles.*

Le problème peut être formalisé comme un "processus de décision markovien" (MDP) où l'on définit :

- *l'état :* S : Position actuelle du robot sur la grille. Un état est une paire $s = (x, y)$ représentant la position dans la grille 5×5.
- *l'action* A : Les actions possibles sont {haut, bas, gauche, droite}.
- *la probabilité de transition* $P(s'|s, a)$: La transition entre les états est déterministe (probabilités égales à 0 ou 1), sauf si l'action mène à un obstacle. Posons $s = (x, y)$, les probabilités de transition peuvent alors être définies de la manière suivante :

$$P(s' \mid s, \text{haut}) = \begin{cases} 1 & \text{si } s' = (x-1, y) \text{ et que } (x-1, y) \text{ est valide} \\ 0 & \text{sinon} \end{cases}$$

De même, pour les autres actions, les transitions sont définies comme suit :

$$P(s' \mid s, \text{bas}) = \begin{cases} 1 & \text{si } s' = (x+1, y) \text{ et que } (x+1, y) \text{ est valide} \\ 0 & \text{sinon} \end{cases}$$

$$P(s' \mid s, \text{gauche}) = \begin{cases} 1 & \text{si } s' = (x, y-1) \text{ et que } (x, y-1) \text{ est valide} \\ 0 & \text{sinon} \end{cases}$$

$$P(s' \mid s, \text{droite}) = \begin{cases} 1 & \text{si } s' = (x, y+1) \text{ et que } (x, y+1) \text{ est valide} \\ 0 & \text{sinon} \end{cases}$$

- *la récompense* $R(s, a)$:
  - $R(s, a) = -1$ si le robot se déplace vers une case vide ou vers un état non optimal.
  - $R(s, a) = +100$ si le robot atteint la destination.
  - $R(s, a) = -10$ si le robot entre en collision avec un obstacle.
- *Objectif* : Maximiser la somme des récompenses cumulées à travers une politique optimale $\pi^*(s)$ décrivant l'ensemble des actions à réaliser.

I.2 Principes de l'apprentissage par renforcement

Principe général

Si l'on exécute une action a dans un état s :

- on obtient une récompense r.
- on arrive dans un nouvel état s'.

En principe, r et s' peuvent dépendre de tout l'historique des états et des actions.

Définition I.2.1. État de Markov

Un état S_t est dit **de Markov** si et seulement si :

$$P[S_{t+1} \mid S_t] = P[S_{t+1} \mid S_1, S_2, \ldots, S_t].$$

Une façon de résumer cela rapidement consiste à dire : "étant donné le présent, le futur est indépendant du passé." Une fois que l'état est connu, on peut ignorer tout l'historique précédent. Par exemple, aux échecs, l'état du jeu (la disposition des pièces sur l'échiquier) ne dépend pas de l'historique des coups joués, mais seulement de l'état actuel du plateau.

On considère en général deux types de modèles :

Observation totale

Dans ce type de modèle, l'agent a une connaissance complète de l'état de l'environnement à chaque instant. Cela signifie qu'il n'y a **aucune incertitude** concernant l'état actuel du monde.

Le cadre théorique utilisé pour ce type de modèle est le **Processus Décisionnel de Markov (PDM)**.

Définition I.2.2. Processus Décisionnel de Markov (PDM)

Un **Processus Décisionnel de Markov** est défini comme un 5-uplet $\langle S, A, T, R, \gamma \rangle$, où :

- S est un **ensemble fini d'états**.
- A est un **ensemble fini d'actions**.
- T est une **matrice de transition** définie par :

$$T^a_{ss'} = P[S_{t+1} = s' \mid S_t = s, A_t = a]$$

  qui représente la probabilité d'arriver dans l'état s' à l'instant $t+1$ en ayant pris l'action a dans l'état s à l'instant t.
- R est un **vecteur de récompenses** défini par :

$$R^a_s = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$$

  qui est la récompense moyenne attendue après avoir pris l'action a dans l'état s.
- $\gamma \in [0,1]$ est un *facteur d'actualisation* ou de *réduction* (en anglais, *discount factor*) qui contrôle l'importance des récompenses futures.

L'objectif est de maximiser la somme des récompenses G_t au fil du temps, qui peut être exprimée par la formule suivante :

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$$

où $\gamma \in [0,1]$ est un facteur de réduction, représentant l'importance relative des récompenses futures par rapport aux récompenses immédiates.

Remarque I.2.3 Facteur de réduction γ

Le facteur de réduction γ reflète l'équilibre que l'agent établit entre les récompenses immédiates et futures. Une valeur de $\gamma = 0$ rend l'agent *myope*, focalisé uniquement sur les gains instantanés, tandis qu'une valeur comprise entre 0 et 1 permet de prendre en compte les récompenses futures tout en leur attribuant une importance décroissante. Cet ajustement est crucial pour modéliser des décisions où le présent et le futur doivent être équilibrés. Lorsque la séquence des récompenses est bornée, cette pondération garantit que la valeur totale des récompenses reste bien définie.

Observation partielle

Dans ce cas, l'agent est **incertain** et ne connaît pas avec certitude l'état actuel du monde. Cependant, il dispose des informations suivantes :

- L'ensemble des états possibles du monde.
- Une estimation probabiliste de la probabilité d'être dans chaque état.

Le cadre théorique correspondant est le **Processus Décisionnel de Markov Partiellement Observable (POMDP)**.

Politique de l'Agent

L'agent suit une *politique* $\pi(a|s)$, qui détermine la probabilité d'exécuter une action a dans un état s. L'objectif de l'agent est d'apprendre la politique π qui maximise l'espérance de la récompense cumulative :

$$V(s_0) = \mathbb{E}[G_t | s_t = s_0]$$

Cela revient à maximiser la *fonction valeur* $V(s)$, qui est la valeur attendue de l'état s sous la politique optimale.

En utilisant le principe de *programmation dynamique*, nous montrerons que la fonction valeur V satisfait une équation de Bellman, du type :

$$V(s) = \max_{a \in A} \big(R(s,a) + \gamma \mathbb{E}[V(s') | s, a] \big)$$

Ainsi, l'agent cherche à trouver la politique optimale $\pi^*(a|s)$ qui maximise $V(s_0)$. En supposant que les récompenses $R(s_0, a_1)$ et $R(s_0, a_2)$ suivent des distributions normales, l'agent pourrait utiliser une approche d'échantillonnage (par exemple, l'algorithme *Upper Confidence Bound* ou UCB) pour estimer la meilleure action à choisir à chaque étape.

I.3 Plan du cours

Ce cours est divisé en deux parties interconnectées :

- Dans la première partie, nous introduirons la notion de contrôle optimal et les principes de la programmation dynamique.
- Dans la seconde partie, nous présenterons les concepts probabilistes fondamentaux de l'apprentissage par renforcement. Cela nous permettra d'aborder quelques algorithmes importants tels que les bandits et le Q-learning.

Voici un tableau résumant les correspondances entre les notions de la théorie du contrôle optimal et celles de la théorie de l'apprentissage par renforcement :

Contrôle optimal	**Apprentissage par renforcement**
État du système	État (s)
Commande ou contrôle	Action (a)
Dynamique du système	Probabilité de transition ($P(s' \mid s, a)$)
Coût	Récompense (r)
Stratégie de contrôle	Politique ($\pi(a \mid s)$)

TABLE I.1 : *Correspondances de vocabulaire entre contrôle optimal et apprentissage par renforcement.*

Remarque I.3.1 Lien avec l'IA

L'apprentissage par renforcement (Reinforcement Learning ou RL) est une branche de l'intelligence artificielle où un agent apprend à prendre des décisions en interagissant avec un environnement. L'agent explore différentes actions, reçoit des récompenses ou des punitions, et ajuste ses comportements pour maximiser les récompenses à long terme. RL est essentiel dans des domaines comme la robotique, les jeux et la gestion autonome. Il permet à l'IA d'apprendre de manière autonome, sans supervision directe, en optimisant une politique de décision. En combinant RL avec l'apprentissage profond, l'IA peut résoudre des problèmes complexes et dynamiques.

II.

Notions de programmation dynamique discrète et continue

II.1 Introduction et motivation

II.1.1 Problèmes de contrôle optimal

Nous avons présenté le problème du *rendez-vous spatial*, un problème de commande optimale, dans la section I.1. Donnons ci-après la forme générale des problèmes de commande optimale que nous allosn considérer dans ce cours.

Système dynamique continu. Soit $T > 0$ donné et U, un compact non vide de $\mathbb{R}^m$. Pour tout $t \in [0,T]$, l'état du système satisfait l'équation d'évolution :

$$x'(t) = g(x(s),u(t),t),$$

L'objectif est de résoudre :

$$\min_{\substack{u:\mathbb{R}\to\mathbb{R} \\ x'(t)=g(t,x(t),u(t))}} \int_0^T f(s,x(s),u(s))\,ds + h(x(T)) \qquad (\mathscr{P}_{\text{cont}})$$

Définition II.1.1. Vocabulaire de l'optimisation dynamique

$f : \mathbb{R}\times\mathbb{R}^n\times\mathbb{R}^m \to \mathbb{R}$ est appelée fonction de coût instantanée
$h : \mathbb{R}^n \to \mathbb{R}$ est appelée fonction de coût final
$g : \mathbb{R}\times\mathbb{R}^n\times\mathbb{R}^m \to \mathbb{R}$ loi de commande
$u(t) \in U \subset \mathbb{R}^m$: ensemble des commandes admissibles

Les fonctions f et h sont supposées de classe C^1.
Voici un de problème typique de commande optimale continue.

Exemple II.1.2

Un consommateur a une durée de vie T. Il gagne un salaire avec un taux $\alpha > 0$ constant (par unité de temps). Soit $x(t)$ son salaire accumulé et $i > 0$ le taux de rémunération (fixe) de l'épargne ou le taux d'intérêt de la dette. La consommation est notée $u(t)$. On a :

$$x'(t) = \alpha + ix(t) - u(t),$$

et $x(0) = x(T) = 0$ (il n'y a pas d"héritage ni de legs). On souhaite déterminer la consommation $u(\cdot)$ qui maximise la fonction d'utilité :

$$u \mapsto \int_0^T \ln u(t)\, e^{-\rho t}\, dt$$

La difficulté ici réside dans le fait que la commande $u(\cdot)$ ne peut pas être choisie de n'importe quelle façon. En effet, seules les commandes $u(\cdot)$ telles que le salaire accumulé $x(\cdot)$ satisfait l'équation dynamique (différentielle)

ainsi que les conditions initiale et terminale $x(0) = x(T) = 0$ doivent être considérées. D'une certaine façon, il est nécessaire de considérer des contraintes "égalité" d'un type très particulier (contrainte différentielle). c'est l'objet de ce chapitre.

Imaginons que l'on cherche à présent à discrétiser le problème ci-dessus pour le résoudre numériquement. On introduit $(t_n)_{0\leq n\leq N}$, une discrétisation uniforme de $[0,T]$ de pas Δt, autrement dit :

$$t_0 = 0, \quad t_i = i\Delta t,\ i \in [\![0,N]\!], \quad t_N = T = N\Delta t.$$

Enfin, on note u_n une approximation de $u(t_n)$, x_n, l'approximation de $x_u(t_n)$ par le schéma d'Euler explicite. Alors, $(x_n)_{0\leq n\leq N}$ satisfait

$$\begin{cases} x_{n+1} = x_n + \Delta t g(t_n, x_n, u_n) & n \in [\![0,N-1]\!] \\ x(0) = x_0, \end{cases}$$

Approchons à présent la fonctionnelle à minimiser à l'aide d'une méthode des rectangles. On obtient :

$$\int_0^T f(s, x_u(s), u(s))ds + h(x(T)) \simeq \Delta t \sum_{n=0}^{N-1} f(t_n, x_n, u_n) + h(x_N).$$

En combinant les approximations ci-dessus, il semble raisonnable d'un point de vue formel de s'intéresser au problème de contrôle optimal discret

$$\inf_{\substack{(u_n)_{0\leq n\leq N}\in U^{N+1} \\ x_{n+1}=x_n+\Delta t g(t_n,x_n,u_n) \\ x(0)=x_0}} \Delta t \sum_{n=0}^{N-1} f(t_n, x_n, u_n) + h(x_N).$$

Bien sûr, ce point de vue est totalement formel et démontrer que les solutions du problèmes discret approchent convenablement en un sens à définir celles du problème continu n'est pas tâche aisée. Il existe une littérature abondante sur ce sujet. Nous renvoyons par exemple à [1, 2].

Dans ce chapitre, nous allons nous intéresser à ces systèmes de contrôle discrets et continus.

Système dynamique discret. Voici le problème général que nous allons aborder dans ce chapitre. Pour unifier la présentation avec le cas continu, nous indexerons les suites par la lettre t et $T \in \mathbb{N}^*$ désignera à présent un entier.

A tout instant $t = 0, 1, 2, \ldots, T$, on définit l'état du système $x(t)$. Il satisfait l'équation d'évolution :

$$\begin{cases} x(0) = x_0, \\ x(t+1) = g(t, x(t), u(t)), & t \in [\![0,T-1]\!] \end{cases}$$

où $\{u(t)\}_{0\leq t\leq T-1}$ désigne la suite de commandes du système discret ci-dessus.

L'objectif est de déterminer la commande $\boldsymbol{u} = (u(0), \ldots, u(T-1))$ à appliquer afin de minimiser le coût total.

$$J(\boldsymbol{u}) = \sum_{t=0}^{T-1} f\big(t, x(t), u(t)\big) + h(x(T)).$$

où f est le coût instantané et h est le coût final.

Le problème s'écrit donc :

$$\boxed{\begin{array}{ll} \min & J(\boldsymbol{u}) \\ u(0),\ldots,u(T-1)\in U & \\ x(t+1)=g(t,x(t),u(t)) & \\ x(0)=x_0 & \end{array}} \qquad (\mathscr{P}_{\text{discr}})$$

Remarque II.1.3

Dans cette partie, nous ne nous préoccuperons pas de l'existence ou l'unicité de solutions aux problèmes traités. En effet, aborder de telles questions nécessiterait des connaissances avancées sur les espaces vectoriels normés (topologie faible). Nous avons fait le choix de nous concentrer sur la caractérisation des solutions à l'aide des conditions d'optimalité.

II.1.2 Plus court chemin sur un graphe

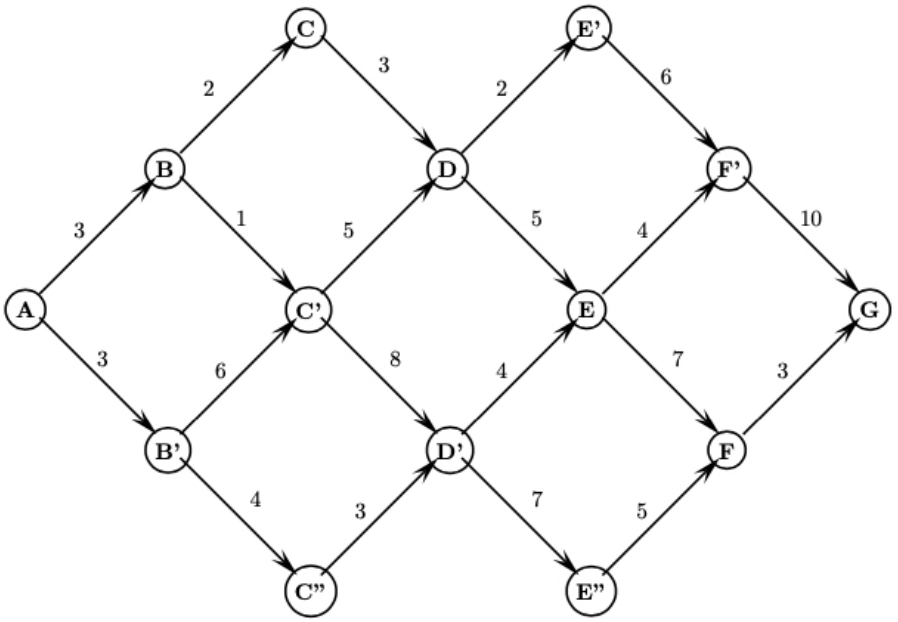
Afin de mieux comprendre la philosophie des approches que nous allons étudier, nous allons de prime abord nous pencher sur la détermination de chemins optimaux sur un graphe, utilisant le principe d'optimalité de Bellman.

Cette approche appelée "programmation dynamique" est une méthode de recherche d'une solution optimale dans un ensemble fini mais très grand. Il s'agit d'une méthode d'énumération implicite : on retient ou rejette des sous-ensembles de solutions mais on ne construit pas toutes les solutions. On rejette certaines solutions sans les avoir construites explicitement si elles appartiennent à un sous-ensemble qui n'est pas intéressant

Exemple II.1.4 Exercice : problème du plus court chemin

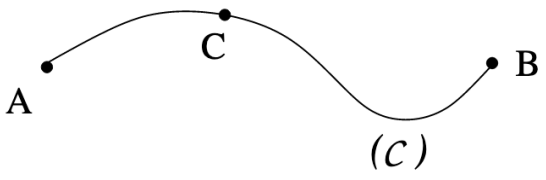
Si le plus court chemin pour aller de A à G passe par D alors le sous chemin allant de D à G est encore un plus court chemin.

Nous allons utiliser ce principe pour déterminer le plus court chemin dans le graphe ci-dessous.



Proposition II.1.5. Principe d'optimalité de Bellman

Un chemin optimal est formé de sous-chemins optimaux : Si $(\mathscr{C})$ est un chemin optimal allant de A à B et si C appartient à $(\mathscr{C})$ alors les sous-chemins de $(\mathscr{C})$ allant de A à C et de C à B sont optimaux.



Par conséquent, une suite de commande est optimale si, quel que soit l'état intermédiaire $x(s)$ pour $s \in [\![0, T-1]\!]$, les commandes ultérieures pour $t \in [\![s, T-1]\!]$ sont optimales pour le sous problème partant de $(x(s), s)$.

Résolution du problème du sous chemin On introduit $L(P)$ = longueur du plus court chemin pour aller de P à G.

- temps $t = 5$.

$L(F) = 3$	FG
$L(F') = 10$	$F'G$

- temps $t = 4$

$L(E') = d(E',F') + L(F') = 16$	$E'F'G$
$L(E'') = d(E'',F) + L(F) = 8$	$E''F'G$
$L(E) = \min(d(E,F') + L(F'), d(E,F) + L(F)) = 10$	EFG

d'après le principe d'optimalité, le sous chemin $EF'G$ ne fait pas partie du chemin minimal entre A et G.

- temps $t = 3$

$L(D) = \min(d(D,E') + L(E'), d(D,E) + L(E)) = \min(18,15) = 15$	$DEFG$
$L(D') = \min(d(D',E) + L(E), d(D',E'') + L(E'')) = \min(14,15) = 14$	$D'EFG$

- temps $t = 2$

$L(C) = d(C,D) + L(D) = 18$	$CDEFG$
$L(C') = \min(d(C',D) + L(D), d(C',D') + L(D')) = \min(20,22) = 20$	$C'DEFG$
$L(C'') = d(C,D') + L(D') = 17$	$C"D'EFG$

- temps $t = 1$

$L(B) = \min(d(B,C) + L(C), d(B,C') + L(C')) = \min(20,21) = 20$	$BCDEFG$
$L(B') = \min(d(B',C') + L(C'), d(B',C'') + L(C'')) = \min(26,21) = 21$	$B'C"D'EFG$

- temps $t = 0$

$L(A) = \min(d(A,B) + L(B), d(A,B') + L(B')) = \min(23,24) = 23$	ABCDEFG

On remarque que l'on n'a pas calculé la longueur de tous les chemins. On en a éliminé au fur et à mesure.

Remarque II.1.6 Algorithme de Dijkstra

L'algorithme précédent s'applique plus généralement à tout graphe sans circuit (sans boucle). Dans l'exemple précédent, toutes les arrêtes sont positives : on aurait donc pu appliquer l'algorithme de Dijkstra qui trouve les plus courts chemins depuis un sommet source vers tous les autres sommets d'un graphe pondéré, à condition que les poids des arêtes soient positifs. Il initialise les distances à l'infini sauf pour la source (0). À chaque itération, il sélectionne le sommet non encore traité ayant la plus petite distance, met à jour les distances des sommets voisins en fonction des arêtes, puis marque ce sommet comme traité. Ce processus se répète jusqu'à ce que tous les sommets soient traités ou qu'aucune amélioration ne soit possible. Le résultat est une table de distances minimales depuis la source.

II.2 Programmation dynamique discrète déterministe

II.2.1 Horizon fini

On considère le problème de contrôle optimal discret ($\mathscr{P}_{\text{discr}}$).

Définition II.2.1. Fonction valeur

La fonction valeur associée au problème ($\mathscr{P}_{\text{discr}}$) est la fonction $V : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ définie par

$$V(s,y) = \min_{\substack{u(s),\dots,u(T-1)\in U \\ x(t+1)=g(t,x(t),u(t)) \\ x(s)=y}} \sum_{t=s}^{T-1} f(t,x(t),u(t)) + h(x(T)).$$

Notons que calculer la valeur optimale du problème ($\mathscr{P}_{\text{discr}}$) revient à calculer $V(0,x_0)$.

Proposition II.2.2. Equation de Hamilton Jacobi Bellman

La fonction valeur satisfait l'équation fonctionnelle :

$$\begin{aligned} V(s,y) &= \min_{u(s)\in U} \left\{ f(s,y,u(s)) + V\left(s+1, g(s,y,u(s))\right)\right\} \\ V(T,y) &= h(y) \end{aligned}$$

Démonstration. L'argument principal est que le critère est additif le long des trajectoires. L'équation s'obtient par simple manipulation des inf. On a :

$$\begin{aligned} V(s,y) &= \min_{\substack{u(s),\dots,u(T-1)\in U \\ x(t+1)=g(t,x(t),u(t)) \\ x(s)=y}} \left(f\left(s,y,u(s)\right) + \sum_{t=s+1}^{T-1} f\left(t,x(t),u(t)\right) + h(x(T)) \right) \\ &= \min_{u(s)\in U} \left(f\left(s,y,u(s)\right) + \min_{\substack{u(s+1),\dots,u(T-1)\in U \\ x(t+1)=g(t,x(t),u(t)) \\ x(s+1)=g(s,y,u(s))}} \sum_{t=s+1}^{T-1} f\left(t,x(t),u(t)\right) + h(x(T)) \right) \\ &= \min_{u(s)\in U} \left(f\left(s,y,u(s)\right) + V(s+1,x(s+1)) \right). \end{aligned}$$

■

Remarque II.2.3 Boucle ouverte/fermée, rétroaction (feedback)

La résolution des problèmes de minimisation ci-dessus fournit de manière rétrograde les vecteurs $u(0)$, ..., $u(T-1)$ comme des rétroactions (feedbacks en anglais) sur l'état x, c'est-à-dire que la commande $u(t)$ est uniquement fonction de l'état $x(t)$ pour chaque valeur de $t \in [\![0, T-1]\!]$.. Une fois qu'on a déterminé tous ces feedbacks en remontant jusqu'à la fonction valeur $V(0,\cdot)$, on est en mesure de déterminer la trajectoire optimale et les contrôles optimaux en repartant de $n=0$.

Plus généralkement, un contrôle *en boucle ouverte* se caractérise par l'absence de rétroaction (feedback). Cela signifie que le système effectue une action selon des instructions prédéfinies, indépendantes de l'état et sans vérifier si l'objectif est atteint. Par exemple, un four programmé pour chauffer pendant 30 minutes le fera, qu'il ait atteint la température souhaitée ou non. En revanche, un problème en boucle fermée intègre une rétroaction. Le système mesure les résultats en temps réel et ajuste son comportement pour corriger les écarts. Un thermostat est un bon exemple : il régule la température en activant ou désactivant le chauffage selon la différence entre la température réelle et la température cible. La notion de feedback désigne donc les informations renvoyées au système pour lui permettre de s'auto-corriger ou de s'adapter. Les boucles fermées sont essentielles pour assurer précision et efficacité dans de nombreux systèmes.

Exemple II.2.4

Soit $x(t)$ la quantité de gaz polluant produit par une usine. L'usine réduit sa production de $u(t)$ unités dans la période $[\![t,t+1]\!]$ On a donc :

$$\begin{aligned} x(t+1) &= x(t) - u(t) \\ x(0) &= x_0 \end{aligned}$$

La taxe de pollution et la réduction de la production ont un coût :

$$\sum_{t=0}^{T-1} \frac{1}{2}(x^2(t) + u^2(t)) + \frac{1}{2}x^2(T)$$

L'équation de Hamilton Jacobi Bellman s'écrit :

$$V(s,y) = \min_{u\in\mathbb{R}} \left\{ \frac{1}{2}(y^2+u^2) + V(y-u, s+1) \right\},$$
$$V(T,y) = \frac{1}{2}y^2$$

Exemple avec $T=3$.

- temps $t=3$. $V(3,y) = \frac{1}{2}y^2$
- temps $t=2$.

$$V(2,y) = \min_{u\in\mathbb{R}} \left\{ \frac{1}{2}(y^2+u^2) + \frac{1}{2}(y-u)^2 \right\} = \min_{u\in\mathbb{R}} \{u^2 - yu + y^2\} = \frac{3}{4}y^2$$

  minimum atteint avec $u(2) = y/2$

- temps $t=1$.

$$V(1,y) = \min_{u\in\mathbb{R}} \left\{ \frac{1}{2}(y^2+u^2) + \frac{3}{4}(y-u)^2 \right\} = \min_{u\in\mathbb{R}} \left\{ \frac{5}{4}y^2 + \frac{5}{4}u^2 - \frac{3}{2}yu \right\} = \frac{4}{5}y^2$$

  minimum atteint avec $u(1) = 3y/5$

- temps $t=0$.

$$V(0,y) = \min_{u\in\mathbb{R}} \left\{ \frac{1}{2}(y^2+u^2) + \frac{4}{5}(y-u)^2 \right\} = \min_{u\in\mathbb{R}} \left\{ \frac{13}{10}y^2 + \frac{13}{10}u^2 - \frac{8}{5}yu \right\} = \frac{21}{26}y^2$$

  minimum atteint avec $u(0) = 8y/13$

Par conséquent, on a :

$$\min_{u(0),u(1),u(2)} \left\{ \sum_{t=0}^{2} \frac{1}{2}(x^2(t)+u^2(t)) + \frac{1}{2}x^2(3) \right\} = \frac{21}{26}x_0^2$$

On a donc :

$$u(0) = \frac{8}{13}x(0), \quad u(1) = \frac{3}{5}x(1), \quad u(2) = \frac{1}{2}x(2),$$

puis :

$$x(1) = x(0) - u(0) = \frac{5}{13}x(0), \quad u(1) = \frac{3}{13}x(0),$$
$$x(2) = x(1) - u(1) = \frac{2}{13}x(0), \quad u(1) = \frac{1}{13}x(0),$$
$$x(3) = x(2) - u(2) = \frac{1}{13}x(0), \quad u(2) = \frac{1}{13}x(0).$$

II.2.2 Horizon infini

Dans cette partie, nous allons considérer et étudier la version à horizon infini du problème standard d'optimisation dynamique discrète :

$$\min_{\substack{u_t\in U\subset\mathbb{R}^m \\ x_{t+1}=g(x_t,u_t) \\ x_0 \text{ donné}}} \sum_{t=0}^{\infty} \beta^t f(x_t,u_t) \tag{II.1}$$

Le nombre β est un facteur de réduction, ce qui signifie que $0<\beta<1$. Comme ni f ni g ne dépendent explicitement de t, ce problème est appelé autonome. Une suite (x_t,u_t) est dite admissible si chaque u_t appartient à U. L'état initial du système est x_0 et l'équation aux différences $x_{t+1} = g(x_t,u_t)$ est satisfaite pour tout $t=0,1,\ldots$.

Comme précédemment, nous ne discuterons pas de l'existence de solutions afin de nous concentrer sur leur caractérisation.

Définition II.2.5. Fonction valeur

La fonction valeur associée au problème (II.1) est la fonction $V : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ définie par

$$V(s,y) = \min_{\substack{u_t \in U \subset \mathbb{R}^m \\ x_{t+1} = g(x_t,u_t) \\ x_s = y}} \sum_{t=s}^{\infty} \beta^t f(x_t,u_t).$$

On a la caractérisation suivante des commandes optimales :

Proposition II.2.6. Equation de Hamilton Jacobi Bellman, horizon infini

On a $V(s,y) = \beta^s V(0,y)$ pour tout s. De plus, en posant $V(y) := V(0,y)$, la fonction V résout l'équation (implicite) :

$$V(x) = \min_{u \in U} \{ f(x,u) + \beta V(g(x,u)) \}.$$

Démonstration. Observons que, puisque f et g sont autonomes (au sens où elles ne dépendent de t que par l'intermédiaire de $u(t)$ et $x(t)$),

$$\begin{aligned} V(s,y) &= \beta^s \min_{\substack{u_t \in U \subset \mathbb{R}^m \\ x_{t+1} = g(x_t,u_t) \\ x_s = y}} \sum_{t=s}^{\infty} \beta^{t-s} f(x_t,u_t) \\ &= \beta^s \min_{\substack{u_t \in U \subset \mathbb{R}^m \\ \tilde{x}_{t+1} = g(\tilde{x}_t,\tilde{u}_t) \\ \tilde{x}_0 = y}} \sum_{t=0}^{\infty} \beta^t f(\tilde{x}_t,\tilde{u}_t) = \beta^s V(0,y), \end{aligned}$$

où l'on a posé $\tilde{x}_t = x_{t-s}$ et $\tilde{u}_t = u_{t-s}$.

Écrivons comme dans le cas d'un horizon fini :

$$\begin{aligned} \beta^s V(s,x) &= \min_{u_s,\ldots \in U} \left(\beta^s f(x,u_s) + \sum_{t=s+1}^{+\infty} \beta^t f(x_t,u_t) \right) \\ &= \min_{u_s \in U} \left(\beta^s f(x,u_s) + \min_{\substack{u_{s+1},\ldots \in U \\ x_{t+1} = g(x_t,u_t) \\ x_{s+1} = g(x,u_s)}} \sum_{t=s+1}^{+\infty} \beta^t f(x_t,u_t) \right) \\ &= \min_{u_s \in U} \left(\beta^s f(x,u_s) + V(s+1, g(x,u_s)) \right) \\ &= \min_{u_s \in U} \left(\beta^s f(x,u_s) + \beta^{s+1} V(0, g(x,u_s)) \right). \end{aligned}$$

Puisque $\beta^s V(s,x) = \beta^s V(x)$ pour tout s, on obtient l'équation attendue en simplifiant par β^s. ■

Exemple II.2.7

Considérons le problème suivant :

$$\max_{\substack{u_t \in \mathbb{R} \\ x_{t+1} = x_t + u_t \\ x_0 \text{ donné}}} \sum_{t=0}^{\infty} \beta^t \left(\frac{2}{3} x_t^2 + u_t^2 \right)$$

Nous avons $f(x,u) = \frac{2}{3}x^2 + u^2$ et $g(x,u) = x + u$. L'équation de Bellman du problème est :

$$V(x) = \min_{u \in \mathbb{R}} \left(\frac{2}{3} x^2 + u^2 + \beta V(x+u) \right).$$

Que faire maintenant ? Comment pouvons-nous trouver une fonction V avec cette propriété ? Nous devinons (de manière ingénieuse) le type de V et essayons $V(x) = \alpha x^2$ pour un $\alpha \in \mathbb{R}$ approprié. Nous obtenons l'équation suivante :

$$\alpha x^2 = \min_{u\in\mathbb{R}} \left(\frac{2}{3}x^2 + u^2 + \alpha\beta(x+u)^2 \right).$$

Un calcul élémentaire fournit que le min est atteint en u tel que

$$u + \alpha\beta(x+u) = 0 \quad \Longleftrightarrow \quad u = -\frac{\alpha\beta}{1+\alpha\beta}x,$$

et ainsi, l'équation de Hamilton Jacobi Bellman devient :

$$\alpha x^2 = \frac{2}{3}x^2 + \left(\frac{\alpha\beta}{1+\alpha\beta}x\right)^2 + \frac{\alpha\beta}{(1+\alpha\beta)^2}x^2$$

ce qui suggère de choisir α solution de l'équation polynomiale de degré 3 à coefficients réels (qui possède donc nécessairement une solution dans $\mathbb{R}$) :

$$\left(\alpha - \frac{2}{3}\right)(1+\alpha\beta)^2 = \left(\alpha\beta\right)^2 + \alpha\beta$$

La proposition II.2.6 suggère un algorithme de détermination de la fonction valeur.

Remarque II.2.8 Vers un algorithme de RL

Dans la deuxième partie de ce cours, dédiée à l'apprentissage par renforcement, nous serons amenés à considérer un algorithme d'approximation de la fonction valeur. Donnons-en le principe, à partir d'une relation de Bellman du type

$$\forall x \in X, \qquad V(x) = \min_{u\in U}\{f(x,u) + \beta V(g(x,u))\}$$

ou du type

$$\forall x \in X, \qquad V(x) = \max_{u\in U}\{f(x,u) + \beta V(g(x,u))\}$$

qui est équivalent au premier, en remplaçant V par $-V$ et f par $-f$.

Étudions (sans perte de généralité) ce dernier problème. On considère l'hypothèse (forte) que l'espace des états X (i.e. l'ensemble des valeurs prises par x) et l'espace des actions/contrôles U sont finis. Alors, on propose l'algorithme suivant, dit d'itération de la valeur :

Données: On se donne V^0. Soient $N^{\max} \in \mathbb{N}^*$ et $\varepsilon > 0$
On pose $\mathrm{Res} = 1$ et $k = 0$;
Tant que $\mathrm{Res} \geq \varepsilon$ *et* $k \leq N^{\max}$ **faire**
  On calcule V^{k+1} en résolvant le problème d'optimisation

$$\forall x \in X, \qquad V^{k+1}(x) = \max_{u\in U}\{f(x,u) + \beta V^k(g(x,u))\}$$

  On pose $\mathrm{Res} = \|V_{k+1} - V_k\|$ et $k \leftarrow k+1$
fin

Algorithme 1 : algorithme d'itération de la valeur (IV).

La convergence est assurée dès que $\gamma \in]0,1[$. En effet, puisque X est fini, il suffit de démontrer le résultat pour un x donné. On appelle $\mathscr{F}$ l'espace des fonctions définies sur X et à valeurs réelles, muni de la norme

$$\|f\|_\infty = \max_{x\in X} |f(x)|.$$

On laisse au lecteur le soin de vérifier que $\|\cdot\|_\infty$ défini une norme sur l'espace vectoriel $\mathscr{F}$ qui le rend complet (on parle d'espace de Banach). Posons

$$T : \mathscr{F} \ni V \longmapsto x \mapsto \max_{u\in U}\{f(x,u) + \beta V(g(x,u))\} \in \mathscr{F}.$$

Pour V et W dans $\mathscr{F}$, on a [1] :

$$\begin{aligned} |TV(x) - TW(x)| &= \left| \max_{u\in U}\{f(x,u) + \beta V(g(x,u))\} - \max_{u\in U}\{f(x,u) + \beta W(g(x,u))\right| \\ &= \leq \beta \max_{u\in U} |V(g(x,u)) - W(g(x,u))| \leq \beta \|V - W\|_\infty. \end{aligned}$$

Puisque $\beta \in]0,1[$, il vient que l'application T est contractante et on en déduit l'existence et l'unicité d'un point fixe $V(x)$, en vertu du théorème du point fixe de Banach [2].

On reviendra sur ce type d'approche dans la suite du cours (Q-learning).

II.3 Programmation dynamique continue déterministe

II.3.1 Le principe du maximum de Pontryagin (PMP)

Définition II.3.1. Problème dynamique en temps continu

Trouver $x(t), u(t)$ qui réalisent le minimum du problème suivant :

$$\min_{\substack{x(t),u(t) \\ x'(t)=g\left(x(t),u(t),t\right) \\ x(0)=\alpha, x(T)=\beta}} \int_0^T f(s,x(s),u(s))\, ds \qquad (\mathscr{P}_{\text{cont}})$$

Définition II.3.2. Hamiltonien

On définit le Hamiltonien du problème ($\mathscr{P}_{\text{cont}}$) :

$$H : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \ni (x,\lambda,u,t) = f(t,x,u) + \langle \lambda, g(t,x,u)\rangle_{\mathbb{R}^n}.$$

1. On utilise que

$$\left| \max_{u\in U} F(u) - \max_{u\in U} G(u) \right| \leq \max_{u\in U} |F(u) - G(u)|.$$

En effet, cette inégalité se récrit sans valeur absolue

$$\max_{u\in U} G(u) \leq \max_{u\in U} |F(u) - G(u)| + \max_{u\in U} F(u) \quad \text{et} \quad \max_{u\in U} F(u) \leq \max_{u\in U} |F(u) - G(u)| + \max_{u\in U} G(u)$$

Démontrons la première, la seconde s'obtenant en intervertissant les rôles joués par F et G. On écrit $G(u) = (G-F)(u) + F(u) \leq |F(u) - G(u)| + F(u)$ puis on passe au max en utilisant que le max de la somme est inférieur à la somme des max.

2. *Théorème du point fixe de Banach-Picard :* soit (E,d) un espace métrique complet, et $f : E \to E$ une application contractante, c'est-à-dire qu'il existe $k \in [0,1[$ tel que, pour tout $(x,y) \in E^2$, $d(f(x),f(y)) \leq k d(x,y)$. Alors f possède un unique point fixe ℓ. De plus, toute suite $(u_n)_{n\in\mathbb{N}}$ définie par $u_0 \in E$ et $u_{n+1} = f(u_n)$, $n \geq 0$, converge vers ℓ.

Proposition II.3.3. Principe du maximum de Pontryagin (PMP)

Soit $(x(\cdot),u(\cdot))$ une solution du problème ($\mathscr{P}_{\text{cont}}$). Alors :

- il existe un état adjoint λ absolument continu, qui satisfait :

$$x'(t) = \partial_\lambda H(t,x(t),u(t),\lambda(t)) =: H_\lambda(t)$$
$$\lambda'(t) = -\partial_x H(t,x(t),u(t),\lambda(t)) =: -H_x(t)$$

  où H_λ (resp. H_x) désigne le gradient de H par rapport à la variable λ (resp. x),

- d'autre part la commande optimale résout le problème

$$\min_{v\in\mathbb{R}^m} H(t,x(t),\lambda(t),v), \qquad \text{(principe d'optimisation instantanée)}$$

  et on a en particulier,

$$\partial_u H(t,x(t),\lambda(t),u(t)) = 0,$$

  donc $u(t)$ est un point critique du Hamiltonien à chaque instant t.

Démonstration. La preuve du PMP est difficile. Nous donnons ci-après une ébauche dans le cas particulier $n = m = 1$. Soit u la commande optimale et x la trajectoire associée. Introduisons le contrôle optimal u (on suppose qu'il existe) et on considère une perturbation $h \in L^2([0,T])$ de u et le contrôle u_ε défini par :

$$u_\varepsilon(t) = u(t) + \varepsilon h(t)$$

En utilisant un argument de type "théorème des fonctions implicites", on peut établir :

$$x_\varepsilon(t) = x(t) + \varepsilon\, \partial_\varepsilon x(t)|_{\varepsilon=0} + \mathrm{o}(\varepsilon),$$

où le terme de reste $\mathrm{o}(\varepsilon)$ peut être considéré uniforme en la variable t. Rappelons que x_ε résout $x'_\varepsilon(t) = g(t,x_\varepsilon(t),u_\varepsilon(t))$. On note $G(\varepsilon)$ la fonction coût évaluée en u_ε, soit

$$G(\varepsilon) = \int_0^T f(t,x_\varepsilon,u_\varepsilon)\,dt.$$

Ainsi, pour toute fonction λ absolument continue, on a :

$$\begin{aligned} G(\varepsilon) &= \int_0^T \big(f(t,x_\varepsilon(t),u_\varepsilon(t)) - \lambda(x'_\varepsilon(t) - g(t,x_\varepsilon(t),u_\varepsilon(t)))\big)\,dt \\ &= \int_0^T \big(f(t,x_\varepsilon(t),u_\varepsilon(t)) + \lambda'(t)x_\varepsilon(t) + \lambda(t)g(t,x_\varepsilon(t),u_\varepsilon(t)))\big)\,dt - [\lambda(t)x_\varepsilon(t)]_0^T \\ &= \int_0^T \big(f(t,x_\varepsilon(t),u_\varepsilon(t)) + \lambda'(t)x_\varepsilon(t) + \lambda(t)g(t,x_\varepsilon(t),u_\varepsilon(t)))\big)\,dt + \lambda(0)\alpha - \lambda(T)\beta \end{aligned}$$

Par hypothèse, G est minimale en $\varepsilon = 0$. On a donc en particulier $G'(0) = 0$, soit :

$$\begin{aligned} G'(\varepsilon) = &\int_0^T \big[\partial_x f(t,x_\varepsilon(t),u_\varepsilon(t)) + \lambda'(t) + \lambda(t)\partial_x g(t,x_\varepsilon(t),u_\varepsilon(t))\big]\partial_\varepsilon x_\varepsilon(t)\,dt \\ &+ \int_0^T \big[\partial_u f(t,x_\varepsilon(t),u_\varepsilon(t)) + \lambda(t)\partial_u g(t,x_\varepsilon(t),u_\varepsilon(t))\big]\partial_\varepsilon u_\varepsilon(t)\,dt \end{aligned}$$

et en $\varepsilon = 0$,

$$\begin{aligned} 0 = G'(0) = &\int_0^T \big[\partial_x f(t,x(t),u(t)) + \lambda'(t) + \lambda(t)\partial_x g(t,x(t),u(t))\big]\,\partial_\varepsilon x_\varepsilon(t)|_{\varepsilon=0}\,dt \\ &+ \int_0^T \big[\partial_u f(t,x(t),u(t)) + \lambda(t)\partial_u g(t,x_\varepsilon(t),u(t))\big]h(t)\,dt \end{aligned}$$

Notons que la formule ci-dessus a été établie indépendamment du choix de la fonction λ et est donc vérifiée quelle que soit λ absolument continue. Afin d'exploiter plus facilement cette formule, on effectue un choix particulier de λ : λ est ainsi supposée solution de l'EDO

$$\lambda'(t) - \lambda(t)\partial_x g(t,x(t),u(t)) + \partial_x f(t,x(t),u(t)) = 0, \tag{II.2}$$

de sorte que

$$0 = G'(0) = \int_0^T \left[\partial_u f(t,x(t),u(t)) + \lambda(t)\partial_u g(t,x_\varepsilon(t),u(t))\right] h(t)\,dt$$

pour tout $h \in L^2([0,T])$. Notons que ce calcul garantit en particulier que le gradient de la fonctionnelle $L^2 \ni u \mapsto$ est donné par

$$u \mapsto \partial_u f(t,x(t),u(t)) + \lambda(t)\partial_u g(t,x_\varepsilon(t),u(t)),$$

où λ est solution de l'EDO (II.2).

Puisque $G'(0) = 0$ quelle que soit h, il vient donc que u vérifie :

$$\partial_u f(t,x(t),u(t)) - \lambda(t)\partial_u g(t,x(t),u(t)) = 0,$$

ce qui revient à dire que $\partial_u H(t,x(t),\lambda(t),u(t)) = 0$ ■

Remarque II.3.4 Fonction valeur et interprétation de l'adjoint

Comme dans le cas discret, introduisons la fonction valeur donnée par

$$V(s,y) = \min_{\substack{u(t) \\ x'(t)=g(x(t),u(t),t) \\ x(s)=y,\ x(T)=\beta}} \int_s^T f(x(t),u(t),t)\,dt.$$

On peut montrer que $\lambda(s) = \partial_y V(y,s)$. Ainsi, $\lambda(s)$ mesure les variations de la fonction coût sur la trajectoire optimale. On dit que $\lambda(s)$ est l'évaluation marginale de la fonction coût à l'instant s : si $x(0) = \alpha$ subit une perturbation de $\delta\alpha$, alors le coût augmentera de $\lambda(0)\delta\alpha$.

Enfin, pour interpréter l'adjoint comme un représentant de Riesz associé à la différentielle de la fonctionnelle de coût, nous renvoyons à l'appendice **??**.

Proposition II.3.5.

Soient f et g convexes et $\mathscr{C}^1$ et soit λ, x, u vérifiant les équations précédentes. Si $\lambda \geqslant 0$ ou si g est affine, alors u est une commande optimale.

Démonstration. Nous donnons la preuve, qui nécessite quelques pré-recquis sur la notion de convexité. Soit (y,v) un autre couple état-controle. Par convexité de f, on a :

$$\begin{aligned}\int_0^T \left(f(t,y(t),v(t)) - f(t,x(t),u(t))\right) dt &\geqslant \int_0^T \left[\partial_x f(t,x(t),u(t))(y(t)-x(t)) + \partial_u f(t,x(t),u(t))(v(t)-u(t))\right] dt \\ &\geqslant \int_0^T \left(-\lambda'(t)(y(t)-x(t)) - \lambda(t)\partial_x g(t,x(t),u(t))(y(t)-x(t))\right) dt \\ &\quad - \int_0^T \lambda(t)\partial_u g(t,x(t),u(t))(v(t)-u(t))\,dt\end{aligned}$$

Or, on a :

$$\begin{aligned}\int_0^T -\lambda'(t)(y(t)-x(t))\,dt &= \left[-\lambda(t)(y(t)-x(t))\right]_0^T + \int_0^T \lambda(y'(t)-x'(t))\,dt \\ &= \int_0^T \lambda(g(t,y(t),v(t)) - g(t,x(t),u(t)))\,dt\end{aligned}$$

D'où :

$$\begin{aligned}\int_0^T f(t,y(t),v(t)) - f(t,x(t),u(t))\,dt \geqslant \int_0^T \lambda(t)\big(&g(t,y(t),v(t)) - g(t,x(t),u(t)) - \partial_x g(t,x(t),u(t))(y(t)-x(t))\big) \\ &- \partial_u g(t,x(t),u(t))(v(t)-u(t))\,dt \geqslant 0,\end{aligned}$$

par convexité de g et positivité de λ ou par linéarité de g. ■

II.3.2 Exemple

Exemple II.3.6 Utilisation du PMP sur un système d'ordre 1

Soit $x(t)$ un stock et $x'(t)$ le taux de production. On souhaite obtenir q produits au temps T en minimisant le coût total :

$$\int_0^T \left(\frac{1}{2}x'(t)^2 + x(t)\right) dt.$$

On reformule ce problème sous la forme de problème de commande optimale :

$$\min_{\substack{x(t),u(t)\\ x'(t)=u(t)\\ x(0)=0,x(T)=q}} \int_0^T \left(\frac{1}{2}u(t)^2 + x(t)\right) dt$$

- Le Hamiltonien est donné par : $H(x,\lambda,u) = \frac{1}{2}u^2 + x + \lambda u$.
- D'après le principe de Pontryagin, si (x,u) est un minimum du problème alors il existe $\lambda : [0,T] \to \mathbb{R}$ telle que

$$\begin{aligned} x' &= u \\ \lambda' &= -H_x = -1 \\ 0 &= H_u = u + \lambda \end{aligned}$$

- Résolution. On a $\lambda(t) = c - t$, puis $u(t) = t - c$. On en déduit que $x'(t) = t - c$ soit encore $x(t) = t^2/2 - ct + a$. On détermine les constantes grâce aux conditions aux limites : $x(0) = 0$ implique $a = 0$ et $x(T) = q$ implique $c = \frac{1}{T}[T^2/2 - q] = T/2 - q/T$.
- Calcul du coût optimal.

$$\int_0^T \frac{1}{2}u(t)^2 + x(t)dt = \int_0^T \frac{1}{2}(t-c)^2 + \frac{t^2}{2} - ctdt = \frac{T^3}{3} + \frac{c^2}{2}t - cT^2 = \frac{qT}{2} + \frac{q^2}{2T} - \frac{T^3}{24}$$

- Le coût d'une production à taux constant (avec $x'(t) = q/T = u(t)$) serait :

$$\int_0^T \frac{1}{2}\frac{q^2}{T^2} + \frac{qt}{T} dt = \frac{q^2}{2T} + \frac{qT}{2}$$

  La différence est de $-T^3/24$. Elle s'amplifie pour T grand.

Exemple II.3.7 Le problème LQ

Considérons un système dynamique linéaire autonome

$$\begin{cases} \dot{x}(t) = Ax(t) + Bv(t) & t \in [0,T], \\ x(0) = x_0 \in \mathbb{R}^n \end{cases} \tag{II.3}$$

associé à la fonction coût quadratique J définie par

$$J(v) = \frac{1}{2}\int_0^T [(Qx(s),x(s)) + (Rv(s),v(s)]\, ds + \frac{1}{2}(Dx(T),x(T))$$

où R est une matrice de $\mathbb{M}_m(\mathbb{R})$ définie positive, $Q \in \mathbb{M}_n(\mathbb{R})$ et $D \in \mathbb{M}_n(\mathbb{R})$ sont supposées semi-définie positives.

Le problème du contrôle optimal LQ s'écrit :

trouver une commande $u \in L^2(0,T;\mathbb{R}^n)$ telle que la trajectoire $x(\cdot)$ associée au système (II.3) minimise J.

Remarquons que dans le modèle considéré, il n'y a pas de contrainte sur les commandes, i.e. $U = \mathbb{R}^m$, et il n'y a pas non plus de contrainte sur l'état final $x(T)$ qui se présente dans la fonction du coût J comme une pénalisation.

On admet :

Théorème II.3.8. Existence et unicité

Le problème

$$\inf_{v \in L^2(0,T,\mathbb{R}^m)} J(v) \tag{II.4}$$

a une solution unique.

Définissons le Hamiltonien H par

$$H : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \ni (x,p,v) = \frac{1}{2}[(Qx,x) + (Rv,v)] + (p, Ax + Bv).$$

Alors le vecteur $v_t := v(t)$ est solution du problème

$$\frac{\partial}{\partial v} H(x(t), p(t), v_t) = 0.$$

Il reste à déterminer le gradient de H par rapport à la variable v. Rappelons le résultat de calcul différentiel :

Lemme II.3.9.

Soit $F : U \subset X \to \mathbb{R}$, une application définie sur un ouvert U d'un espace de Banach X. Si F est différentiable en $x \in X$, alors sa différentielle dans la direction $h \in X$ est donnée par

$$DF(x) \cdot h = \lim_{\varepsilon \searrow 0} \frac{F(x + \varepsilon h) - F(x)}{\varepsilon}.$$

Calculons $\partial_v H(x(t), p(t), v)$. On a :

$$H(x,p,v) = \frac{1}{2}(Rv,v) + (v, B^\top p) + \text{Reste}$$

où le terme Reste ne dépend pas de v. Par conséquent,

$$D_v H(x,p,v) \cdot h = \lim_{\varepsilon \searrow 0} \frac{1}{\varepsilon} \left(\frac{1}{2}(R(v + \varepsilon h), v + \varepsilon h) + (v + \varepsilon h, B^\top p) - \frac{1}{2}(Rv,v) - (v, B^\top p) + \right)$$

Ainsi pour tout $0 \le t \le T$ fixé, la fonction $v \mapsto H(x(t), p(t), v)$ atteint son minimum en $v_t = -R^{-1} B^\top p(t)$. Autrement dit, la commande optimale v_t réalise le minimum instantané de cette fonction.

Ainsi, on caractérise les trajectoires optimales comme suit :

Théorème II.3.10. Principe du maximum de Pontryagin - Problème LQ

La trajectoire x, associée à la commande u, est optimale si et seulement s'il existe une variable d'état adjoint p définie par

$$\begin{cases} \dot{p}(t) + A^\top p(t) = -Qx(t) & t \in [0,T] \\ p(T) = Dx(T). \end{cases} \tag{II.5}$$

De plus la commande optimale u est caractérisée par

$$u(t) = -R^{-1} B^\top p(t) \tag{II.6}$$

Remarque II.3.11 extension du résultat précédent

Le théorème II.3.8 reste valable si $T=\infty$ avec $Q=0$, pourvu que le système (II.3) soit controlable en un certain temps T fini. En effet, s'il existe une commande v sur $[0,T]$ qui ramène x_0 à 0, alors on peut prolonger la commande $v(t)$ par zéro pour $t \geq T$ de telle sorte que la trajectoire $x(t)$ est réduite à $\{0\}$ pour $t \geq T$. On a ainsi trouvé une trajectoire du coût fini et la borne inférieure de J est également finie. On peut alors effectuer le même procédé de minimisation comme dans le cas "$T<\infty$" (Exercice).

II.3.3 Autres contraintes terminales

Définition II.3.12. Problème dynamique en temps continu

Trouver $x(t), u(t)$ qui réalisent le minimum du problème suivant :

$$\min_{\substack{x(t),u(t)\\ x'(t)=g(x(t),u(t),t)\\ x(0)=\alpha,}} \int_0^T f(x(s),u(s),s)\,ds + h(x(T))$$

où $h(x(T)$ est une pénalité ou une prime.

Proposition II.3.13. Condition de transversalité

(i) Si l'extrémité est totalement libre, on impose $\lambda(T) = h'(x(T))$.

(ii) Si l'extrémité est soumise à une contrainte d'inégalité, de la forme $\phi(x(T)) \leqslant 0$, alors on introduit p tel que :

$$\lambda(T) = h'(x(T)) + p\phi'(x(T)), \qquad p \geqslant 0, \quad p\phi(x(T)) = 0.$$

Exemple II.3.14 Exercice

Résoudre le problème

$$\min_{\substack{x(t),u(t)\\ x'(t)=u(t)\\ x(0)=1}} \int_0^T \frac{1}{2}(u(t)^2 + x(t)^2)\,dt$$

- Le Hamiltonien est donné par : $H(x,u,\lambda) = \frac{1}{2}(u^2+x^2) + \lambda u$.
- D'après le principe de Pontryagin, si (x,u) est un minimum du problème alors il existe $\lambda : [0,T] \to \mathbb{R}$ telle que

$$\begin{aligned} x' &= u \\ \lambda' &= -H_x = -x \\ 0 &= H_u = u + \lambda \end{aligned} \qquad\qquad x(0)=1, \lambda(T)=0.$$

- Résolution. On a $x' = -\lambda$ et $x'' = -\lambda' = x$. Donc $x(t) = A\cosh t + B\sinh t$. On détermine les constantes grâce aux conditions limites : $x(0)=1$ implique $A=0$, $\lambda(T) = -x'(T) = 0$ implique $B = -\sinh T/\cosh T$.

On obtient donc :

$$x(t) = \cosh t - \frac{\sinh T}{\cosh T}\sinh t = \frac{\cosh t\cosh T - \sinh T\sinh t}{\cosh T} = \frac{\cosh(T-t)}{\cosh T}$$
$$u(t) = -x'(t) = \frac{\sinh(T-t)}{\cosh T}$$

- Le coût minimal est donné par :

$$\int_0^T \frac{1}{2}\frac{\cosh^2(T-t)+\sinh^2(T-t)}{\cosh^2 T}dt = \int_0^T \frac{1}{2}\frac{\cosh(2(T-t))}{\cosh^2 T}dt = \left[-\frac{\sinh(2(T-t))}{2\cosh^2 T}\right]_0^T$$
$$= \frac{\sinh(2T)}{2\cosh^2 T} = \frac{2\cosh T\sinh T}{2\cosh^2 T} = \tanh T$$

Exemple II.3.15 Exercice

On s'intéresse à présent à une dynamique d'ordre 2 : $x'' = u$. Pour appliquer le PMP, il s'agit de la récrire comme un problème du premier ordre en posant $x' = y$ et $y' = u$.

Résoudre le problème

$$\min_{\substack{x(t),y(t),u(t)\\ x'(t)=y(t)\\ y'(t)=u(t)\\ x(0)=0,y(0)=0\\ x(T)+y(T)\geqslant 2}} \int_0^T \frac{1}{2}u(t)^2 dt$$

- Le Hamiltonien est donné par : $H(x,y,u,\lambda) = \frac{1}{2}u^2 + \lambda_1 y + \lambda_2 u$.
- D'après le principe de Pontryagin, si (x,u) est un minimum du problème alors il existe $\lambda_1, \lambda_2 : [0,T] \to \mathbb{R}$ telle que

$$\begin{aligned}
&x' = y\\
&y' = u\\
&\lambda_1' = -H_x = 0\\
&\lambda_2' = -H_y = -\lambda_1\\
&0 = H_u = u + \lambda_2\\
&x(0) = 0, y(0) = 0, \lambda_1(T) = -p, \lambda_2(T) = -p,\\
&p(x(T) + y(T) - 2) = 0.
\end{aligned}$$

- Résolution. On a $\lambda_1(t) = -p$ et $\lambda_2(t) = pt + a$. De la condition $\lambda_2(T) = -p$, on déduit $\lambda_2(t) = p(t-T-1)$.

  Si $p = 0$, alors $\lambda_2 = 0$ et $u = 0$. On en déduit que $x = 0$ et $y = 0$. Ce qui est impossible.

  Si $p > 0$, alors $x(T)+y(T) = 2$. On a $y'(t) = u(t) = p(T+1-t)$, soit $y(t) = pt(T+1) - pt^2/2$ qui vérifie bien la condition $y(0) = 0$. On en déduit ensuite de la relation $x'(t) = y(t)$, que $x(t) = pt^2/2(T+1) - pt^3/6$, qui vérifie bien la relation $x(0) = 0$.

  Enfin la relation $x(T) + y(T) = 2$ permet d'obtenir : $p(-T^3/6 + T^2/2(T+1) - T^2/2 + T(T+1)) = 2$ soit

$$p = \frac{2}{(T^3/3 + T^2 + T)} = \frac{6}{T(T^2+3T+3)}$$

II.3.4 Ajout de contraintes ponctuelles sur le contrôle

On se donne un sous-ensemble $U \subset \mathbb{R}^m$.

Définition II.3.16. Problème dynamique en temps continu

Trouver $x(t), u(t)$ qui réalisent le minimum du problème suivant :

$$\min_{\substack{x(t),u(t)\in U \\ x'(t)=g(x(t),u(t),t) \\ x(0)=\alpha,}} \int_0^T f(x(s),u(s),s)ds + h(x(T))$$

où $h(x(T)$ est une pénalité ou une prime.

Proposition II.3.17. Condition de maximisation instantanée

Les équations énoncées dans la proposition II.3.3 restent valides, à l'exception de

$$\min_{v\in\mathbb{R}^m} H(t,x(t),\lambda(t),v),$$

qui est remplacée par

$$\min_{v\in U} H(t,x(t),\lambda(t),v).$$

Exemple II.3.18 Pêche optimale

On considère $x(t)$, la quantité d'une certaine espèce de poisson dans un lac à l'instant t. Si on ne fait pas de prélèvement, c'est-à-dire si on ne pêche pas du tout, la population à l'instant t de cette espèce de poisson augmente à taux constant $\alpha > 0$, en suivant la loi d'évolution $\dot{x}(t) = \alpha x(t)$.

Pêcher consiste à prélever une proportion de population, $u(t)x(t)$, où $u(\cdot)$ désigne une application de $\mathbb{R}_+$ dans $\mathbb{R}$ telle que $0 \le u(\cdot) \le u_{\max}$, avec $u_{\max} \in]0,1[$, un taux maximal fixé par les régulations de pêche de l'espèce en question. À l'instant initial, la population de pêche est estimée à $x(0) = x_0 > 0$. L'équation différentielle modélisant l'évolution de $x(t)$, que nous noterons dorénavant $x_u(t)$ pour mettre en valeur la dépendance de cette quantité en la fonction u, s'écrit alors :

$$\begin{cases} \dot{x}_u(t) = \alpha x_u(t) - u(t)x_u(t), & t > 0. \\ x_u(0) = x_0 \end{cases} \tag{II.7}$$

La période de pêche est fixée, $[0,T]$, avec $T > 0$ donné. L'objectif est de maximiser la pêche totalisée durant la période fixée, c'est-à-dire maximiser $\int_0^T u(t)x(t)\,dt$.

Le problème de contrôle optimal s'écrit alors :

$$\sup_{u\in\mathscr{U}_{ad}} J(u), \quad \text{avec} \quad J(u) = \int_0^T x_u(t)u(t)\,dt, \tag{$\mathscr{P}$}$$

où x_u désigne l'unique solution de (II.7) et

$$\mathscr{U}_{ad} = \{u \in L^\infty([0,T]) \mid 0 \le u(t) \le u_{\max} \text{ p.p. dans } [0,T]\}.$$

Le Hamiltonien du problème ($\mathscr{P}$) est donné par :

$$\begin{array}{rccl} \mathscr{H}: & \mathbb{R}\times\mathbb{R}\times\mathbb{R} & \to & \mathbb{R} \\ & (x,\lambda,u) & \mapsto & -xu + \lambda(\alpha x - ux). \end{array}$$

Les conditions d'optimalité fournissent l'existence de λ absolument continu, tel que

$$\begin{aligned} \dot{x} &= \partial_\lambda H = \alpha x - ux \\ \dot{\lambda} &= -\partial_x H = u - \alpha\lambda + u\lambda. \end{aligned}$$

On a de plus la condition de maximisation instantanée : $u(t)$ résout le problème

$$\min_{v\in[0,u_{\min}]} \mathscr{H}(x(t),\lambda(t)v) \text{ équivalent au problème } \min_{v\in[0,u_{\max}]} v(-x(t)-\lambda(t)x(t)).$$

On a la condition de transversalité (voir Proposition II.3.13) $p(T)=0$.

Remarquons que $x(\cdot)>0$. En effet, puisque $x(\cdot)=0$ est une solution particulière de l'équation principale du système, on déduit du théorème d'unicité de Cauchy-Lipschitz que s'il existe un $t_0\in[0,T]$ tel que $x_u(t_0)=0$, alors, $x_u(\cdot)=0$, en contradiction avec la condition initiale.

Résolvons le problème de minimisation instantanée. Notons que $x>0$. On obtient :

$$\begin{aligned}&\text{sur } \{x+\lambda x>0\}=\{\lambda>-1\}, \quad \text{on a } u=u_{\max}\\ &\text{sur } \{x+\lambda x<0\}=\{\lambda<-1\}, \quad \text{on a } u=0.\end{aligned}$$

Puisque λ est continue et que $\lambda(T)=0$, il existe un voisinage de T inclus dans $\{\lambda>-1\}$, et d'après les conditions d'optimalité, on a nécessairement $u=u_{\max}$ sur cet intervalle.

D'après l'équation adjointe, on a $\lambda'(t)=(\lambda(t)+1)u(t)-\alpha\lambda(t)$. Ainsi, si $\lambda(t_0)=-1$, alors $\lambda'(t_0)=\alpha$. On en déduit que $\lambda>-1$ après t_0 et $\lambda<-1$ avant t_0. Ainsi, t_0 est nécessairement isolé et $|\{\lambda=-1\}|=0$. On en déduit que le contrôle optimal u est bang-bang, égal à 0 ou $u_{\max}$ p.p.

Sur un intervalle de l'ensemble $\{u=0\}$, on a $\lambda'=-\alpha\lambda$ et $\lambda\le-1<0$, donc λ est croissante.

On a déjà vu que u est *bang-bang*. Il reste à étudier le nombre de commutations. Supposons l'existence de t_0, un point de commutation de $\{u=u_{\max}\}$ à $\{u=0\}$. Alors, on a vu $\lambda'(t_0)=-\alpha<0$ et ainsi, $\lambda>-1$ après t_0 ce qui signifie que $u=u_{\max}$ après t_0. On obtient une contradiction. Par conséquent, on en déduit qu'il existe au plus un point de commutation.

II.4 Travaux Pratiques : contrôle optimal

Cette séance de TP a pour but la prise en main d'un logiciel "boîte noire" IpOpt[3] (Interior Point OPTimizer) permettant la résolution de problèmes d'optimisation non-linéaires à l'aide d'une méthode dite de points intérieurs. IpOpt sert à trouver un minimum (local) de problèmes d'optimisation de la forme

$$\inf_{x\in K} f(x) \qquad \text{où} \qquad K=\{x\in\mathbb{R}^n \mid g_L\le g(x)\le g_U\},$$

où $f:\mathbb{R}^n\to\mathbb{R}$ et $g:\mathbb{R}^n\to\mathbb{R}^p$ sont des fonctions régulières. IpOpt se révèle souvent efficace, même pour traiter des problèmes de grande dimension. IpOpt est accessible gratuitement par l'intermédiaire du package Python appelé `Gekko`. Si vous utilisez votre machine personnelle, vous pouvez l'installer à l'aide de la ligne de commande `pip install gekko`.

Tout au long de ce TP, vous pourrez vous référer à la page web relative au package `Gekko` :

https://gekko.readthedocs.io/en/latest/index.html

Pour vous aider, le notebook `Start_Gekko.ipynb` contient quelques exemples de résolution de problèmes d'optimisation et de contrôle optimal[4], vous pouvez le télécharger sur la page web du cours.

Préliminaires

Résoudre théoriquement le problème d'optimisation non-linéaire

$$\inf_{(x_1,x_2)\in K} x_1^2+x_2^2-14x_1-6x_2-7 \quad \text{avec} \quad K=\{(x_1,x_2)\in\mathbb{R}^2 \mid x_1+x_2\le 2 \text{ et } x_1+2x_2\le 3\}$$

Utiliser alors `Gekko` pour résoudre numériquement ce problème.

Problèmes de contrôle optimal à résoudre numériquement

3. Pour aller plus loin et comprendre quel algorithme est implémenté, on peut se référer à l'article :
A. Wächter and L. T. Biegler, On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming, Mathematical Programming 106(1), pp. 25-57, 2006
que l'on peut télécharger sur la page web du cours.

4. Ces exemples sont extraits de la page https://apmonitor.com/wiki/index.php/Main/GekkoPythonOptimization.

Problème de temps minimal. (contrôle d'un tram, 1ère version) On veut déterminer le contrôle $u(\cdot)$ et temps minimal T nécessaire pour que la solution (x,y) du système contrôlé

$$\begin{cases} \dot{x}(t) = y(t) & t \in]0,T[\\ \dot{y}(t) = u(t) & t \in]0,T[\\ x(0) = y(0) = 0 \end{cases}$$

satisfasse $x(T) = 0$, $y(T) = -1$, sous la contrainte $|u(t)| \leq 1$ p.p. $t \in]0,T[$. Résoudre numériquement ce problème.

Problème de temps minimal. (contrôle d'un tram, 2ème version). Modifier le problème précédent pour minimiser une combinaison convexe du temps final et de $\frac{1}{2}\int_0^T (\dot{x}(t)^2 + x(t)^2)\,dt$ et pour avoir $x(T) = 0$ et $y(T) \in [-1-\varepsilon, -1+\varepsilon]$ où $\varepsilon > 0$ est un paramètre fixé par l'utilisateur.

Contrôle d'insectes. Pour traiter une population $x(0)$ d'insectes nuisibles, on introduit dans l'écosystème une population $y(0)$ d'insectes prédateurs (non nuisibles), se nourrissant des nuisibles. On suppose que les prédateurs que l'on introduit se reproduisent, de manière proportionnelle au nombre de nuisibles. Le contrôle $u(\cdot)$ est le taux de disparition des prédateurs. Pour simplifier l'écriture on normalise les variables de façon à ce qu'aucune constante biologique n'intervienne dans l'écriture du système. Le modèle s'écrit alors

$$\begin{cases} \dot{x}(t) = x(t)(1 - y(t)) & t \in]0,T[\\ \dot{y}(t) = -y(t)(u(t) - x(t)) & t \in]0,T[\\ x(0) = 1,\ y(0) = 4 \end{cases}$$

où le contrôle $u(t)$ vérifie la contrainte ponctuelle : $0 < 1 \leq u(t) \leq 3$ p.p. $t \in [0,T]$.

Analyse du système. Démontrer que, pour tout contrôle u, $x(t) > 0$ et $y(t) > 0$ sur $[0,T]$.

Contrôle optimal de ce système. On demande de résoudre numériquement le problème de contrôle optimal

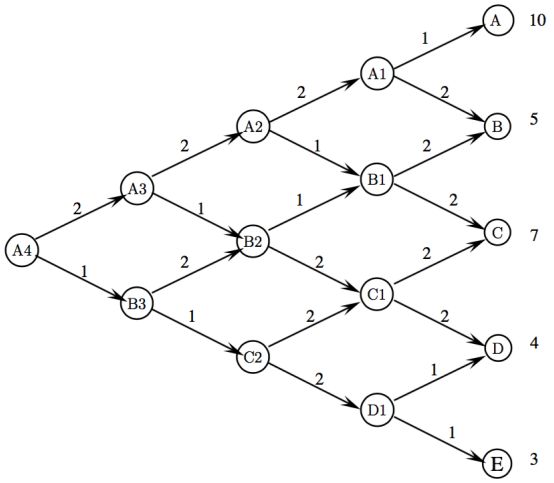
$$\inf_u \{T \mid x(T) = a,\ y(T) = 1\}$$

où $a \in [1,3]$ est donné. Tracer la fonction de contrôle obtenue et la trajectoire optimale associée. Commenter les résultats obtenus.

On souhaite modifier le problème et inclure dans le critère, en plus du temps T, le coût L^2 du contrôle, soit $\int_0^T u(t)^2\,dt$? Modéliser ce problème et le résoudre.

II.5 Exercices du chapitre

II.1 Etant donné 5 villes A, B, C, D et E possédant chacun un trésor (10,5,7,4,3). Le réseau et les coûts de transport sont schématisés sur le graphe ci-dessous. Déterminer le chemin optimal partant de A_4 pour trouver le gain maximal.



II.2 A chaque instant $t = 0,1,2,3$, on dispose des ressources $x(t) \geq 0$ que l'on peut affecter au cours de la période $[t, t+1[$ à l'usage

(i) qui apporte le gain 1 par unité et entraîne un amortissement de 20% des ressources affectées,

(ii) qui apport le gain 2 par unité et entraîne un amortissement de 50% des ressources affectées.

Notons $u(t)$ la quantité de ressources à affecter à l'usage (i) et $x(t) - u(t)$ celle à l'usage (ii). On cherche à maximiser la fonction du gain total :

$$J(u) = \sum_{t=0}^{3} (2x(t) - u(t))$$

sous la contrainte dynamique :

$$x(t+1) = 0.5\,x(t) + 0.3\,u(t), \quad 0 \leq u(t) \leq x(t), \quad x(0) = 1.$$

Déterminer explicitement le gain maximal obtenu, la suite des commandes optimales $u(0)$, $u(1)$, $u(2)$, $u(3)$ et la suite des états correspondants $x(0)$, $x(1)$, $x(2)$, $x(3)$, $x(4)$.

II.3 Soit $x_0 > 0$. On souhaite maximiser la fonction de gain

$$J(x,u) = \sum_{t=0}^{2} \left(1 + x(t) - u(t)^2\right) + x(3),$$

sous la contrainte dynamique $x(t+1) = x(t) + u(t)$, $x(0) = x_0$.

Déterminer :

- la fonction valeur aux temps successifs $t = 3$, $t = 2$, $t = 1$ et $t = 0$.
- la solution optimale $x(0)$, $u(0)$, $x(1)$, $u(1)$, $x(2)$, $u(2)$, $x(3)$ en fonction de x_0.
- le gain associé en fonction de x_0.

II.4 Un consommateur dispose d'un capital initial de 100 euros. Il peut dépenser une partie $u(t)$ de son capital $x(t)$ et épargner le reste avec un taux d'intérêt de 100% :

$$x(t+1) = 2(x(t) - u(t)), \quad x(0) = 100.$$

Déterminer la stratégie optimale qui maximise la fonction d'utilité :

$$\sum_{t=0}^{2} (4u(t) - 0.001\,u^2(t)).$$

II.5 Minimiser la fonction

$$\int_0^1 u^2(t)\,dt + x^2(1)$$

sous la contrainte dynamique :

$$x'(t) = x(t) + u(t), \quad x(0) = 1.$$

II.6 Au cours d'une période $T > 0$, un capital $x(t)$ peut être soit réinvesti avec un taux d'intérêt $\alpha > 0$, soit consommé. La fonction d'utilité s'écrit $\ln u(t)$ où $u(t) > 0$ désigne la quantité consommée. Maximiser la fonction :

$$J(u) = \int_0^T \ln u(t)\,dt$$

sous la contrainte dynamique :

$$x'(t) = \alpha x(t) - u(t), \quad u(t) \geq 0, \quad x(0) = x_0, \quad x(T) \geq 0.$$

II.7 Dans le modèle de Life-cycle saving, supposons que l'individu a eu un héritage $a > 0$ et a décidé de laisser un legs $b > 0$. On maximise donc la fonction d'utilité :

$$J(u) = \int_0^T \ln u(t)\, e^{-\rho t}\, dt$$

sous la contrainte dynamique de l'équation d'état du salaire accumulé :

$$x'(t) = \alpha + i\,x(t) - u(t), \quad x(0) = a, \quad x(T) = b.$$

Quel est le plus grand legs qu'il puisse laisser ?

II.8 On considère un réservoir d'eau dont la hauteur d'eau au temps s est notée $y(s)$ et qui subit une perte d'eau linéaire en temps et auquel on peut ajouter de l'eau au cours du temps. On modélise le système par l'équation différentielle commandée

$$\begin{cases} y'(s) = u(s) - \gamma s, & s \in [0,T] \\ y(0) = 0. \end{cases}$$

où $\gamma > 0$ est une constante donnée, $T > 0$ est l'horizon de temps (fixé), la hauteur d'eau au temps de départ est nulle et la fonction u est la commande modélisant l'ajout d'eau. On suppose que le coût d'ajout d'eau est donné par $\int_0^T u(s)^2\, ds$ et on souhaite qu'au temps final, la hauteur d'eau du réservoir soit la plus proche possible de la hauteur $h > 0$. Pour déterminer une stratégie optimale de remplissage du réservoir, on considère le problème suivant :

$$\text{minimiser } J(u) \text{ définie par } J(u) = \frac{1}{2}\int_0^T u(s)^2\, ds + \frac{M}{2}(y(T) - h)^2$$

où $M > 0$ est une constante fixée.

(i) Énoncer le principe du maximum de Pontryagin pour ce problème, en précisant les conditions de transversalité s'il y a lieu. On appellera λ l'état adjoint associé à ce problème.

(ii) Montrer qu'il existe trois constantes $a < 0$, $b < 0$ et $c \in \mathbb{R}$ que l'on précisera telles que

$$y(s) = as^2 + b(y(T) - h)s + c, \quad s \in [0,T].$$

En déduire $y(T)$ en fonction de a, b et c.

(iii) Calculer $\lim_{M \to +\infty} y(T)$ et interpréter le résultat (on expliquera en particulier pourquoi il est intéressant de considérer des choix de M très grands dans la fonction à minimiser).

II.9 *(problème de la reine Didon)* Soient ℓ et L positifs tels que $2\ell < L < \pi\ell$. On cherche à résoudre le problème :

$$\text{maximiser } \int_{-\ell}^{\ell} x(t)\, dt \text{ par rapport à } u \in C^0([-\ell,\ell]),$$

où (x,y) désigne la solution du système dynamique

$$\begin{cases} \dot{x}(t) = u(t), & t \geq 0 \\ \dot{y}(t) = \sqrt{1 + u(t)^2}, & t \geq 0 \\ x(-\ell) = x(\ell) = 0, & \\ y(-\ell) = 0,\ y(\ell) = L, & \end{cases}$$

On **admet** que ce problème possède une solution.

(i) On définit la fonction $\hat{h}$ sur $\mathbb{R}$ par $\hat{h}(u) = p_1 u + p_2\sqrt{1+u^2}$ où $(p_1,p_2) \in \mathbb{R}^2$. On suppose que le problème $\sup_{\mathbb{R}} \hat{h}$ possède une solution. Montrer que, nécessairement, $p_2 < 0$.

(ii) Caractériser le contrôle optimal. On montrera en particulier :

- qu'il existe $a > 0$ et $b \in \mathbb{R}$ tels que le contrôle optimal u vérifie

$$\frac{u(t)}{\sqrt{1+u(t)^2}} = -\frac{t-b}{a}.$$

- que $(t,x(t))$ se trouve sur une portion de cercle.

III.

Programmation dynamique stochastique

III.1 Problématique générale

Rappelons que, par opposition aux autres types d'apprentissage 'classique' (apprentissage supervisé, non-supervisé), l'objectif de l'apprentissage par renforcement est d'apprendre par une *interaction* avec l'environnement. On veut donc formaliser l'idée que nos décisions, qui sont prises en fonction de l'environnement, ont également, à leur tour, une influence sur celui-ci, positive ou négative, et dont nous pouvons tenir compte pour prendre notre prochaine décision. D'un point de vue formel, et contrairement à un algorithme (statique, 'one shot') d'apprentissage, on prend donc des décisions séquentiellement, en fonction d'un environnement que nous faisons évoluer. Formellement, nous allons apprendre la meilleure stratégie en fonction d'une chaîne de Markov (les états successifs) et non plus simplement d'une famille de v.a. i.i.d. (les données brutes d'un modèle d'apprentissage).

A compléter.

III.2 Processus Décisionnels de Markov

III.2.1 Définition et construction

Nous rappelons et complétons la définition suivante, abordée dans le chapitre introductif.

36

Simple example of MDP: Green Robot

(From Barto y Sutton):

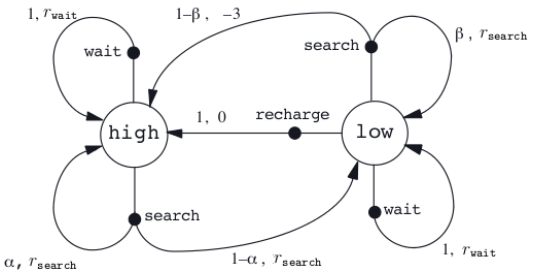


Figure 1: Transitions green robot

Définition III.2.1. Processus Décisionnel de Markov (PDM)

Un **Processus Décisionnel de Markov** est caractérisé par un 5-uplet $\langle \mathscr{S}, \mathscr{A}, T, \mathscr{R}, \gamma \rangle$, où :

- $\mathscr{S}$ est un **ensemble fini d'états**. On note pour tout $t \in \mathbb{N}$ (ou $t \in [\![0,T]\!]$ si l'on joue à l'horizon T), $S_t \in \mathscr{S}$, l'*état* du PDM à l'instant t.
- $\mathscr{A}$ est un **ensemble fini d'actions**. On note pour tout $t \in \mathbb{N}$ (ou $t \in [\![0,T]\!]$), $A_t \in \mathscr{A}$, l'*action* prise à l'instant t.
- T est une **matrice de transition** définie par :

$$T^a_{s \to s'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s, A_t = a\right], \text{ pour tous } s,s' \in \mathscr{S}, a \in \mathscr{A},$$

  qui représente, pour tout instant t, la probabilité d'arriver dans l'état s' à l'instant $t+1$ en ayant pris l'action a dans l'état s à l'instant t. Notons que nous supposons l'*homogénéité* du modèle, i.e., la quantité précédente dépend de a,s et s' mais pas de t.
- $\mathscr{R}$ est l'ensemble des récompenses. Pour tout t, on note $R_t \in \mathscr{R}$, la récompense issue de l'action A_t prise dans l'état S_t. Cette récompense dépend formellement également de l'état S_{t+1}, elle est donc représentée comme une variable aléatoire $R_{t+1} := R_{t+1}(S_t, A_t)$, mesurable par rapport à la tribu $\mathscr{F}_{t+1}$. On définit alors pour tous $s \in \mathscr{S}, a \in \mathscr{A}$, la *récompense moyenne* ayant pris l'action a dans l'état s à l'instant t, par

$$\begin{aligned} \bar{R}_{t+1}(s,a) &= \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right] \\ &= \mathbb{E}\left[R_{t+1}(S_t, A_t) \mid S_t = s, A_t = a\right] \\ &= \mathbb{E}\left[R_{t+1}(s,a)\right]. \end{aligned}$$

- $\gamma \in [0,1]$ est un *facteur d'actualisation* ou de *réduction* (en anglais, *discount factor*) qui contrôle l'importance des récompenses futures.

Construction dynamique

Concrètement, un PDM peut donc être représenté par la suite aléatoire $((S_t, A_t, R_t))_{t \in \mathbb{N}}$ représentant pour tout $t \in \mathbb{N}$ (ou $t \in [\![0,T]\!]$), le triplet état/action/récompense à l'instant t. Celle-ci est construite comme suit :

(i) On part de l'état $S_0 = s \in \mathscr{S}$.

(ii) On prend une action $A_0 = a \in \mathscr{A}$. Celle-ci (mais pas seulement celle-ci, a priori un aléa extérieur intervient aussi) induit le passage vers un état $S_1 \in \mathscr{S}$ à l'instant 1, tiré au sort suivant la loi $T^a_{s,\cdot}$ indépendamment de tout le reste.

(iii) On reçoit une récompense $R_1 \in \mathscr{R}$, dont la valeur moyenne est donnée par $\bar{R}_1(s,a)$.

(iv) On recommence les étapes (i) à (iii) en partant de $S_1 = s' \in \mathscr{S}$.

Exemple III.2.2 Le Robot Vert

Considérons un exemple file-rouge simple : le "Robot Vert", qui circule dans les bureaux d'une entreprise, et qui collecte les déchets qui ont été abandonnés par terre et sur les bureaux. Le robot est chargé par batterie, et quand la batterie est déchargée, on le recharge. Nous allons, pour simplicité, considérer que le Robot est caractérisé uniquement par deux états possibles : l'état *high* (batterie très chargée) et l'état em low (batterie peu chargée). On considère également que :

(i) Dans l'état *high*, le robot a deux actions possibles :

(ia) *search* : Se déplacer pour chercher activement des déchets. Ceci "rapporte" en moyenne plus de déchets, mais cela coûte de la batterie. Nous considérons alors que :

- La probabilité que la batterie soit *low* à l'instant suivant est de $1-\alpha$ et donc, la probabilité qu'elle reste *high* est α.
- La récompense moyenne (comptée en quantités de déchets récoltés) est alors de r_{search}.

(ib) *wait* : Le robot peut décider de rester en place là où il se trouve. Nous considérons alors que, dans cet état :

- Le robot ne collecte que les déchets qui lui sont apportés. La récompense moyenne est alors de $r_{\text{wait}} < r_{\text{search}}$.
- Le robot ne se décharge pas.

(ii) Dans l'état *low*, le robot a trois actions possibles :

(iia) *search* : Se déplacer pour chercher activement des déchets. Ceci rapporte toujours en moyenne une récompense de r_{search} par unité de temps. Par ailleurs :

- Ave probabilité β, le robot reste dans l'état *low* à l'instant suivant ; sa batterie n'est pas encore déchargée.
- En revanche, avec probabilité $1-\beta$, la batterie du robot se décharge. On porte alors le robot à la recharge, ce qui correspond à une récompense négative de $-r_{\text{load}}$. A l'instant suivant, le robot est alors chargé totalement, soit dans l'état *high*.

(iib) *wait* : Le robot peut décider de rester en place, là où il se trouve. Comme dans le cas *high*, il reste alors dans l'état *low* à l'instant suivant, et perçoit une récompense moyenne de r_{wait}.

(iic) *recharge* : Le robot peut décider seul de se rendre à la base de recharge. A l'instant suivant, il sera donc toalement chargé, dans l'état *high*. Cette opération aboutit à une récompense nulle.

Les états, les actions et les probabilités de transitions sont donnée en Figure 1, pour $r_{\text{load}} = 3$. On peut en particulier, donner les probabilités de transitions étant donné l'état : les transitions s'écrivent

$$\begin{array}{llll}
T^{\text{search}}_{\text{high}\to\text{low}} = 1-\alpha; & T^{\text{search}}_{\text{high}\to\text{high}} = \alpha \\
T^{\text{wait}}_{\text{high}\to\text{low}} = 0; & T^{\text{wait}}_{\text{high}\to\text{high}} = 1 \\
T^{\text{search}}_{\text{low}\to\text{high}} = 1-\beta; & T^{\text{search}}_{\text{low}\to\text{low}} = \beta \\
T^{\text{wait}}_{\text{low}\to\text{high}} = 0; & T^{\text{wait}}_{\text{low}\to\text{low}} = 1 \\
T^{\text{recharge}}_{\text{low}\to\text{high}} = 1; & T^{\text{recharge}}_{\text{low}\to\text{low}} = 0.
\end{array}$$

III.2.2 Politiques de décision

Dans la construction précédente, il reste à préciser comment sont déterminées, à chaque instant, les actions en fonction de l'état (et d'autre chose ?). D'où les définitions suivantes.

Définition III.2.3. Règle de décision markovienne

Pour tout $t \in \mathbb{N}$ (ou $t \in [\![0,T]\!]$), une **règle de décision markovienne** à l'instant t est la donnée des lois de probabilités conditionnelles $\pi_t(.|s), s \in \mathscr{S}$ sur $\mathscr{A}$, où pour tout $a \in \mathscr{A}$,

$$\pi_t(a|s) = \mathbb{P}[A_t = a \,|\, S_t = s],$$

c'est-à-dire, la loi de probabilité de A_t sachant que $S_t = s$. En d'autres termes, l'action à t ne dépend que de l'état et d'un aléa exogène. On dit que la règle de décision markovienne à t est **déterministe** si pour tout $s \in \mathscr{S}$, il existe $a(s) \in \mathscr{A}$ tel que $\pi_t(a(s)|s) = 1$. En d'autres termes, l'action à t est entièrement déterminée par l'état.

Définition III.2.4. Politique markovienne

Une **politique markovienne** Π est la donnée d'une famille de règles de décision markoviennes $\pi_t(.|s), s \in \mathscr{S}, t \in \mathbb{N}$ (ou $t \in [\![0,T]\!]$). Une politique markovienne est dite **déterministe** si toutes les règles de décision le sont. Elle est dite **stationnaire** si les règles de décision sont constantes au fil du temps, i.e., $\pi_t = \pi$ pour tout $t \in \mathbb{N}$ (ou $t \in [\![0,T]\!]$).

Exemple III.2.5 Le Robot Vert (suite)

Revenons à l'exemple du "Robot Vert". À ce stade, aucune hypothèse n'est faite sur la politique. Par exemple, elle est :

(i) **markovienne**, mais **non stationnaire**, si dans chaque état *low, high*, on fait un tirage au sort parmi les actions *search, wait, reload* suivant une loi de probabilité qui dépend de l'instant ;

(ii) **markovienne** et **stationnaire**, si dans chaque état, on fait un tirage au sort parmi les actions suivant une loi de probabilité qui ne dépend pas de l'instant ;

(iii) **déterministe**, si dans l'un (ou l'autre) des cas précédent, la loi de probabilité est une Dirac en une action a, qui ne dépend que de l'état courant (et éventuellement de l'instant).

Dans la suite, on notera $\mathbb{P}_\Pi$ la mesure de probabilité sous la politique markovienne π, et $\mathbb{E}_\Pi$, l'espérance correspondante.

Proposition III.2.6. Propriété de Markov

Pour toute politique markovienne Π fixée, la suite $(S_t)_{t\in\mathbb{N}}$ est une chaîne de Markov. Si Π est stationnaire, cette chaîne de Markov est homogène.

Démonstration. Pour tout $t \in \mathbb{N}$ (ou $t \in [\![0,T]\!]$) et tous $s,s' \in \mathscr{S}$, on a

$$\begin{aligned}
&\mathbb{P}_\Pi\left[S_{t+1}=s' \,|\, S_t=s, S_{t-1}=s_{t-1}, \cdots, S_0=s_0\right] \\
&\quad= \sum_{a\in\mathscr{A}} \mathbb{P}_\Pi\left[S_{t+1}=s', A_t=a \,|\, S_t=s, S_{t-1}=s_{t-1}, \cdots, S_0=s_0\right] \\
&\quad= \sum_{a\in\mathscr{A}} \mathbb{P}_\Pi\left[S_{t+1}=s' \,|\, A_t=a, S_t=s, S_{t-1}=s_{t-1}, \cdots, S_0=s_0\right] \mathbb{P}_\Pi\left[A_t=a \,|\, S_t=s, S_{t-1}=s_{t-1}, \cdots, S_0=s_0\right] \\
&\quad= \sum_{a\in\mathscr{A}} T^a_{ss'}\pi_t(a|s) \\
&\quad= \sum_{a\in\mathscr{A}} \mathbb{P}_\Pi\left[S_{t+1}=s' \,|\, A_t=a, S_t=s\right] \mathbb{P}_\Pi\left[A_t=a \,|\, S_t=s\right] \\
&\quad= \mathbb{P}_\Pi\left[S_{t+1}=s' \,|\, S_t=s\right].
\end{aligned}$$

Cette dernière quantité vaut

$$\sum_{a\in\mathscr{A}} T^a_{s\to s'}\pi(a|s)$$

dans le cas stationnaire, une probabilité de transition qui ne dépend donc pas de t. ■

On peut alors démontrer de la même façon que

Proposition III.2.7. Propriété de Markov état/action

Pour toute politique markovienne Π fixée, la suite $((S_t,A_T))_{t\in\mathbb{N}}$ est une chaîne de Markov à valeurs dans $\mathscr{S}\times\mathscr{A}$. Si Π est stationnaire, cette chaîne de Markov est homogène.

Définition III.2.8. Transitions de $(S_t)_{t\in\mathbb{N}}$ - cas stationnaire

Si la politique markovienne Π est stationnaire, on a vu que le processus $(S_t)_{t\in\mathbb{N}}$ est une chaîne de Markov homogène. on notera pour tous $s,s' \in \mathscr{S}$,

$$\begin{aligned} P_\Pi(s,s') &= \mathbb{P}_\Pi\left[S_{t+1} = s' \mid S_t = s\right] \\ &= \sum_{a\in\mathscr{A}} \mathbb{P}\left[S_{t+1} = s' \mid S_t = s; A_t = a\right] \pi(a|s) \\ &= \sum_{a\in\mathscr{A}} T^a_{s\to s'} \pi(a|s). \end{aligned}$$

Si $\mathscr{S}$ est fini, P_Π est donc une matrice carrée de taille $|\mathscr{S}|$, markovienne.

Définition III.2.9.

Soit Π, une politique markovienne. La **récompense moyenne** à t **étant donnée** Π, est la valeur moyenne de la récompense à l'instant t sachant seulement l'état s, c'est à dire, la quantité

$$\begin{aligned} \bar{R}_{\Pi,t}(s) &= \mathbb{E}_\Pi\left[\bar{R}_{t+1}(s,a)\right] \\ &= \sum_{a\in\mathscr{A}} \bar{R}_{t+1}(s,a)\mathbb{P}_\Pi\left[A_t = a \mid S_t = s\right] \\ &= \sum_{a\in\mathscr{A}} \bar{R}_{t+1}(s,a)\pi_t(a \mid s). \end{aligned}$$

Cette quantité ne dépend pas de t en particulier si Π est stationnaire, on la note alors

$$\bar{R}_\Pi(s) := \sum_{a\in\mathscr{A}} \bar{R}_1(s,a)\pi(a \mid s).$$

III.2.3 Fonctions de valeurs et caractérisations

Soit un facteur de réduction $\gamma \in [0,1]$. Rappelons alors la forme de la fonction objectif : pour tout $t \in \mathbb{N}$ (ou $t \in [\![0,T]\!]$),

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+1+k},$$

où γ est le facteur d'actualisation.

Définition III.2.10. Fonctions de valeur

Pour tout $t \in \mathbb{N}$ (ou $t \in [\![0,T]\!]$), la **fonction de valeur** (ou fonction de valeur état) à t est définie par l'espérance de la récompense cumulative en fonction de l'état initial, sous la politique Π. Elle s'écrit donc

$$v_{\Pi,t} : \begin{cases} \mathscr{S} & \longrightarrow \mathbb{R}; \\ s & \longmapsto \mathbb{E}_\Pi [G_t | S_t = s]. \end{cases}$$

La fonction **fonction de valeur état/action** à t est définie de manière équivalente, en conditionnant également par l'action prise à l'instant t. Elle s'écrit donc

$$\tilde{v}_{\Pi,t} : \begin{cases} \mathscr{S} \times \mathscr{A} & \longrightarrow \mathbb{R}; \\ (s,a) & \longmapsto \mathbb{E}_\Pi [G_t | S_t = s; A_t = a], \end{cases}$$

Si la chaîne de Markov $(S_t)_{t\in\mathbb{N}}$ est homogène, ce qui est le cas en particulier si la politique markovienne est stationnaire, la quantité précédente ne dépend pas de t et les fonctions de valeur s'écrivent respectivement

$$v_\Pi : \begin{cases} \mathscr{S} & \longrightarrow \mathbb{R}; \\ s & \longmapsto \mathbb{E}_\Pi [G_0 | S_0 = s]; \end{cases} \tag{III.1}$$

$$\tilde{v}_\Pi : \begin{cases} \mathscr{S} \times \mathscr{A} & \longrightarrow \mathbb{R}; \\ (s,a) & \longmapsto \mathbb{E}_\Pi [G_0 | S_0 = s; A_0 = a]. \end{cases} \tag{III.2}$$

Proposition III.2.11. Caractérisation de la fonction de valeur

Soit $\gamma \in [0,1)$, et l'opérateur L_Π, défini sur $\mathscr{C}(\mathscr{S},\mathbb{R})$ par

$$L_\Pi : \begin{cases} \mathscr{C}(\mathscr{S},\mathbb{R}) & \longrightarrow \mathscr{C}(\mathscr{S},\mathbb{R}) \\ v & \longmapsto L_\Pi v : \begin{cases} \mathscr{S} & \to \mathbb{R} \\ s & \mapsto \bar{R}_\Pi(s) + \gamma \mathbb{E}_\Pi [v(S_{t+1}) | S_t = s]. \end{cases} \end{cases} \tag{III.3}$$

Alors, pour toute politique markovienne stationnaire Π, la fonction de valeur v_Π définie par (III.1) satisfait l'équation de point fixe

$$v_\Pi = L_\Pi v_\Pi. \tag{III.4}$$

Démonstration. On a pour tout t, presque-sûrement,

$$\begin{aligned} G_t &= R_{t+1} + \sum_{k=1}^{\infty} \gamma^k R_{t+1+k} \\ &= R_{t+1} + \gamma \sum_{k=1}^{\infty} \gamma^{k-1} R_{t+2+k-1} \\ &= R_{t+1} + \gamma \sum_{\ell=0}^{\infty} \gamma^{\ell} R_{t+2+\ell} \\ &= R_{t+1} + \gamma G_{t+1}. \end{aligned} \tag{III.5}$$

Par conséquent, pour tout $s \in \mathscr{S}$ et pour tout $t \in \mathbb{N}$ on a

$$\begin{aligned}
v_\Pi(s) &= \mathbb{E}_\Pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\Pi[R_{t+1} \mid S_t = s] + \gamma \mathbb{E}_\Pi[G_{t+1} \mid S_t = s] \\
&= \mathbb{E}_\Pi[R_{t+1}(S_t, A_t) \mid S_t = s] + \gamma \sum_{s' \in \mathscr{S}} \mathbb{E}_\Pi\left[G_{t+1} \mid S_{t+1} = s';\, S_t = s\right] \mathbb{P}\left[S_{t+1} = s' | S_t = s\right] \\
&= \sum_{a \in \mathscr{A}} \mathbb{E}[R_{t+1}(S_t, A_t) \mid S_t = s;\, A_t = a] \mathbb{P}_\Pi[A_t = a | S_t = s] + \gamma \sum_{s' \in \mathscr{S}} \mathbb{E}_\Pi\left[G_{t+1} \mid S_{t+1} = s'\right] \mathbb{P}\left[S_{t+1} = s' | S_t = s\right] \\
&= \sum_{a \in \mathscr{A}} \mathbb{E}[R_{t+1}(s,a) \mid S_t = s;\, A_t = a] \pi_t(a|s) + \gamma \sum_{s' \in \mathscr{S}} v_\Pi(s') \mathbb{P}\left[S_{t+1} = s' | S_t = s\right] \\
&= \sum_{a \in \mathscr{A}} \bar{R}_1(s,a) \pi(a|s) + \gamma \mathbb{E}_\Pi[v_\Pi(S_{t+1}) | S_t = s] \\
&= \bar{R}_\Pi(s) + \gamma \mathbb{E}_\Pi[v_\Pi(S_{t+1}) | S_t = s] \\
&= L_\Pi v_\Pi(s),
\end{aligned}$$

où l'on utilise la propriété de Markov dans la quatrième égalité, et la stationnarité de la politique dans la sixième. Cette égalité pour tout $s \in \mathscr{S}$ signifie exactement que la fonction de valeur v_Π satisfait à l'équation de point fixe $v_\Pi = L_\Pi v_\Pi$. ■

Corollaire III.2.12. Calcul de la fonction de valeur

Si $\mathscr{S}$ est fini, pour tout Π stationnaire il existe une unique fonction de valeurs v_Π.

Démonstration. Comme la politique est stationnaire, d'après (III.3) et (III.4) le vecteur $|\mathscr{S}|$-dimensionnel $(v_\Pi(s), s \in \mathscr{S})$ s'écrit comme solution du système suivant :

$$\begin{aligned}
v_\Pi(s) &= \bar{R}_\Pi(s) + \gamma \sum_{s' \in \mathscr{S}} v_\Pi(s') \mathbb{P}_\Pi\left[S_1 = s' \mid S_0 = s\right] \\
&= \bar{R}_\Pi(s) + \gamma \sum_{s' \in \mathscr{S}} v_\Pi(s') P_\Pi(s,s') \\
&= \bar{R}_\Pi(s) + \gamma P_\Pi v_\Pi(s).
\end{aligned}$$

en rappelant la définition III.2.8. En d'autres termes, en ordonnant l'ensemble $\mathscr{S}$ et en écrivant les vecteurs considérés en colonne, nous obtenons que v_Π s'écrit comme une solution du système linéaire

$$v_\Pi = \bar{R}_\Pi + \gamma P_\Pi v_\Pi,$$

ou encore,

$$(I - \gamma P_\Pi) v_\Pi = \bar{R}_\Pi. \tag{III.6}$$

Remarquons maintenant que

$$(I - \gamma P_\Pi) \left(\sum_{i=0}^{+\infty} \gamma^i (P_\Pi)^i \right) = I,$$

où la somme précédente a un sens puisque $0 < \gamma < 1$ et chaque matrice $(P_\Pi)^i$, $i \in \mathbb{N}$, est stochastique. La matrice $I - \gamma$ est donc inversible, et son inverse est précisément la matrice définie par cette dernière somme. Finalement, v_Π correspond donc au vecteur colonne

$$v_\Pi = (I - \gamma P_\Pi)^{-1} \bar{R}_\Pi.$$

■

Comme le résultat précédent le suggère, ayant fixé une politique stationnaire Π, il "suffit" de résoudre un système linéaire. De fait, cette tâche est en général infaisable en pratique, dès lors que l'espace d'états $\mathscr{S}$ est assez grand. D'où l'utilité du résultat suivant. Dans la suite, on note

Corollaire III.2.13. Convergence vers la fonction de valeur

Pour toute fonction $v \in \mathscr{C}(\mathscr{S},\mathbb{R})$, le schéma itératif

$$\begin{cases} v_0 & = v; \\ v_{n+1} & = L_\Pi v_n, \quad n \in \mathbb{N} \end{cases}$$

converge vers l'unique solution du système (III.4).

Démonstration. D'après le Théorème du point fixe, il suffit de montrer que l'application L_Π est γ-contractante, c'est à dire,

$$\| L_\Pi(v_2) - L_\Pi(v_1) \|_\infty \leq \| v_2 - v_1 \|_\infty, \tag{III.7}$$

où pour toute fonction $v : \mathscr{S} \to \mathbb{R}$,

$$\| v \|_\infty = \max_{s \in \mathscr{S}} |v(s)|.$$

Mais ceci découle immédiatement de ce que

$$\begin{aligned} \| L_\Pi(v_2) - L_\Pi(v_1) \|_\infty &= \max_{s \in \mathscr{S}} \left| \left\{ \bar{R}_\Pi(s) + \gamma \mathbb{E}_\Pi [v_2(S_1) | S_0 = s] \right\} - \left\{ \bar{R}_\Pi(s) + \gamma \mathbb{E}_\Pi [v_1(S_1) | S_0 = s] \right\} \right| \\ &= \max_{s \in \mathscr{S}} \gamma \left| \mathbb{E}_\Pi [v_2(S_1) - v_1(S_1) | S_0 = s] \right| \\ &\leq \max_{s \in \mathscr{S}} \gamma \mathbb{E}_\Pi [\| v_2 - v_1 \|_\infty | S_0 = s] \\ &= \max_{s \in \mathscr{S}} \gamma \| v_2 - v_1 \|_\infty = \gamma \| v_2 - v_1 \|_\infty . \end{aligned}$$

■

La convergence du schéma itératif proposé ci-dessus vers la fonction de valeur v_Π induit donc l'algorithme suivant pour l'estimation de v_Π,

Données: On se donne v'. Soit $\varepsilon > 0$
On pose Res = 1 et $k = 0$;
Tant que Res $\geq \varepsilon$ **faire**
  Pour tout $s \in \mathscr{S}$ Faire
  Poser

$$v(s) = v'(s)$$

  Calculer

$$v'(s) = \sum_{a \in \mathscr{A}} \left\{ \bar{R}_1(s,a) + \gamma \sum_{s' \in \mathscr{S}} v(s') T^a_{s \to s'} \right\} \pi(a|s)$$

  Poser Res = $\| v' - v \|$
fin

Algorithme 2 : Estimation de la fonction de valeur.

Proposition III.2.14. Caractérisation de la fonction de valeur état/action

Soit $\gamma \in [0,1)$, et l'opérateur $\tilde{L}_\Pi$, défini sur $\mathscr{C}(\mathscr{S},\mathbb{R})$ par

$$\tilde{L}_\Pi : \begin{cases} \mathscr{C}(\mathscr{S} \times \mathscr{A},\mathbb{R}) & \longrightarrow \mathscr{C}(\mathscr{S} \times \mathscr{A},\mathbb{R}) \\ v & \longmapsto \tilde{L}_\Pi v : \begin{cases} \mathscr{S} \times \mathscr{A} & \to \mathbb{R} \\ (s,a) & \mapsto \bar{R}_1(s,a) + \gamma \mathbb{E}_\Pi [v(S_1,A_1) | S_0 = s; A_0 = a]. \end{cases} \end{cases} \tag{III.8}$$

Alors, pour toute politique markovienne stationnaire Π, la fonction de valeur état/action $\tilde{v}_\Pi$ définie par (III.2) satisfait l'équation de point fixe

$$\tilde{v}_\Pi = \tilde{L}_\Pi \tilde{v}_\Pi.$$

Démonstration. En appliquant à nouveau (III.5), il vient pour tout $s \in \mathscr{S}$ et $a \in \mathscr{A}$, pour tout t,

$$\begin{aligned}
\tilde{v}_\Pi(s,a) &= \mathbb{E}_\Pi[G_t \mid S_t = s; A_t = a] \\
&= \mathbb{E}_\Pi[R_{t+1} \mid S_t = s; A_t = a] + \gamma \mathbb{E}_\Pi[G_{t+1} \mid S_t = s; A_t = a] \\
&= \mathbb{E}[R_{t+1}(S_t, A_t) \mid S_t = s; A_t = a] \\
&\qquad + \gamma \sum_{s' \in \mathscr{S}} \sum_{a' \in \mathscr{S}} \mathbb{E}_\Pi\left[G_{t+1} \mid S_{t+1} = s'; A_{t+1} = a'; S_t = s; A_t = a\right] \mathbb{P}\left[S_{t+1} = s'; A_{t+1} = a' | S_t = s; A_t = a\right] \\
&= \mathbb{E}[R_{t+1}(s,a) \mid S_t = s; A_t = a] + \gamma \sum_{s' \in \mathscr{S}} \sum_{a' \in \mathscr{S}} \mathbb{E}_\Pi\left[G_{t+1} \mid S_{t+1} = s'; A_{t+1} = a'\right] \mathbb{P}\left[S_{t+1} = s'; A_{t+1} = a' | S_t = s; A_t = a\right] \\
&= \mathbb{E}[R_1(s,a) \mid S_t = s; A_t = a] + \gamma \sum_{s' \in \mathscr{S}} \sum_{a' \in \mathscr{S}} \tilde{v}_\Pi(s',a') \mathbb{P}\left[S_{t+1} = s'; A_{t+1} = a' | S_t = s; A_t = a\right] \\
&= \bar{R}_1(s,a) + \gamma \mathbb{E}_\Pi[\tilde{v}_\Pi(S_{t+1}, A_{t+1}) | S_t = s; A_t = a] \\
&= \bar{R}_1(s,a) + \gamma \mathbb{E}_\Pi[\tilde{v}_\Pi(S_1, A_1) | S_0 = s; A_0 = a] \\
&= \tilde{L}_\Pi \tilde{v}_\Pi(s,a),
\end{aligned}$$

où l'on utilise à nouveau la propriété de Markov dans la quatrième égalité, et l'homogénéité dans l'avant-dernière. ■

Cas d'un état absorbant

Supposons que la suite $(S_t)_{t \in \mathbb{N}}$ admette des états absorbants, i.e., il existe un sous-ensemble d'état $\mathscr{S}' \subset \mathscr{S}$ tels que pour tout $s' \in \mathscr{S}'$,

$$\text{Pour tout } t,\ \left[S_t = s'\right] \Longrightarrow \left[\text{Pour tout } t' \geq t, S_{t'} = s'\right].$$

Dans ce cas, on considère que toutes les actions partant d'un état $s' \in \mathscr{S}'$ sont toutes équivalentes (puisqu'elles mènent au même état), et que la récompense obtenue à chaque nouvelle étape après l'atteinte de l'état absorbant, est nulle. alors il existe une application

$$r : \begin{cases} \mathscr{S}' & \longrightarrow \mathscr{S}' \\ s' & \longmapsto r(s') = \mathbb{E}\left[\bar{R}_1(s',a)\right] = r(s'), \end{cases}$$

qui représente la récompense lorsque l'on atteint l'état s' (et que l'on y reste). Dans ce cas, on peut naturellement étendre la définition de la fonction de valeur de la manière suivante.

Proposition III.2.15. Fonctions de valeur - cas d'un état absorbant

Soit $\gamma \in [0,1)$, et les opérateurs L_Π et $\tilde{L}_\Pi$ définis par (III.3) et (III.8), respectivement. Alors, s'il existe un sous-ensemble $\mathscr{S}'$ d'état absorbant, les fonctions de valeur satisfont pour toute Π markovienne et stationnaire, pour tout $s \in \mathscr{S}$,

$$v_\Pi(s) = \begin{cases} L_\Pi v_\Pi(s), & \text{si } s \in \mathscr{S} \setminus \mathscr{S}'; \\ r(s) & \text{si } s \in \mathscr{S}', \end{cases}$$

et pour tous $s \in \mathscr{S}$, $a \in \mathscr{A}$,

$$\tilde{v}_\Pi(s,a) = \begin{cases} \tilde{L}_\Pi \tilde{v}_\Pi(s,a), & \text{si } s \in \mathscr{S} \setminus \mathscr{S}'; \\ r(s) & \text{si } s \in \mathscr{S}'. \end{cases}$$

Démonstration. Il suffit de remarquer que dans ce cas, pour tout $s' \in \mathscr{S}'$,

$$v_\Pi(s') = \bar{R}_\Pi(s') + \gamma \mathbb{E}_\Pi[v_\Pi(S_1) | S_0 = s] = r(s'),$$

et de même, pour tout $a \in \mathscr{A}$, comme aucune action n'est prise dans l'état s',

$$\tilde{v}_\Pi(s',a) = \bar{R}_1(s',a) + \gamma \mathbb{E}_\Pi\left[\tilde{v}_\Pi(S_1) | S_0 = s'; A_0 = a\right] = r(s').$$

■

Exemple III.2.16 Le dilemme de l'étudiant

On considère maintenant le "dilemme de l'étudiant", représenté par la figure Figure 2. Les états de l'étudiant sont représentés par l'ensemble $\mathscr{S} := [\![1,7]\!]$, et les actions, par l'ensemble $\mathscr{A} :=$ {travailler,se reposer}. Les transitions sont représentées en Figure 2.

44

Example: Student dilemma

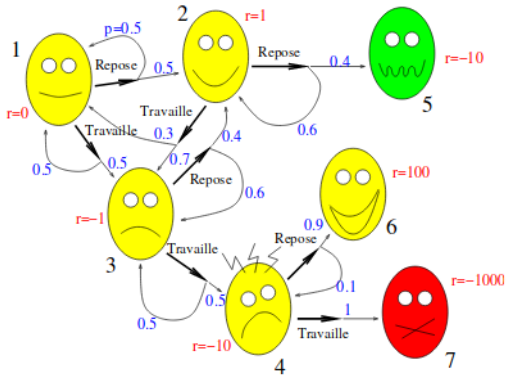


Figure 2: Image R. Munoz.

III.2.4 Politique optimale

Il est naturel de se poser la question suivante : peut-on construire une politique markovienne (éventuellement stationnaire) *optimale*, au sens ou elle maximise la fonction valeur en tout point? Comme on va le voir, il existe une réponse simple, quoiqu'*a priori* seulement théorique, à cette question.

Définition III.2.17. Politique optimale

On dit qu'une politique Π^* est **optimale**, si l'on a pour tout $s \in \mathscr{S}$,

$$v_{\Pi^*}(s) = \max_{\Pi} v_{\Pi}(s). \tag{III.9}$$

Dans cette définition, on peut restreindre la notion d'optimalité à un sous-ensemble $\mathscr{P}$ de politiques, par exemple, les politiques markoviennes, les politiques markoviennes stationnaires, les politiques constantes, etc.

Pour déterminer une politique Π^* satisfaisant (III.9) pour tout s, il s'agit donc de maximiser l'espérance

d'une somme aléatoire p.s. convergente, ce qui est *a priori* faisable par un calcul exact, si les ensembles $\mathscr{S}$ et $\mathscr{A}$ sont finis. Cependant, ces ensembles peuvent en général être très grands, ce qui rend ces calculs infaisables en pratique. Il faut alors recourir à d'autres techniques, et possiblement à des approximations. Nous démontrons ci-après que la fonction de valeur optimale v_{Π^*} satisfait, elle aussi, une équation de point fixe. Définissons tout d'abord l'opérateur d'optimalité de Bellman.

Définition III.2.18. Opérateur d'optimalité de Bellman

L'opérateur d'optimalité de Bellman L^* sur $\mathscr{C}(\mathscr{S},\mathbb{R})$ est défini par

$$L^* : \begin{cases} \mathscr{C}(\mathscr{S},\mathbb{R}) & \longrightarrow \mathscr{C}(\mathscr{S},\mathbb{R}) \\ v & \longmapsto L^*v : \begin{cases} \mathscr{S} & \longrightarrow \mathbb{R} \\ s & \longmapsto \max_{a\in\mathscr{A}} \left\{ \bar{R}_1(s,a) + \gamma \mathbb{E}\left[v(S_1)|S_0 = s; A_0 = a\right] \right\}. \end{cases} \end{cases} \tag{III.10}$$

En d'autres termes, d'après la formule de transfert on a pour tout $v : \mathscr{S} \to \mathbb{R}$ et tout $s \in \mathscr{S}$,

$$L^*v(s) = \max_{a\in\mathscr{A}} \left\{ \bar{R}_1(s,a) + \gamma \sum_{s'\in\mathscr{S}} v(s') T^a_{s\to s'} \right\}.$$

Remarquons le résultat suivant,

Proposition III.2.19. Unicité - Equation de Bellman

L'équation de point fixe de Bellman

$$v^* = L^*v^*, \quad v^* \in \mathscr{C}(\mathscr{S},\mathbb{R}), \tag{III.11}$$

admet une unique solution v^* appelée *fonction de valeur optimale* vers laquelle, pour toute $v \in \mathscr{C}(\mathscr{S},\mathbb{R})$, le schéma itératif de puissance itérée

$$\begin{cases} v_0 & = v; \\ v_{n+1} & = L^*v_n, \quad n \in \mathbb{N} \end{cases}$$

converge pour la norme infinie.

Démonstration. Remarquons que l'application L^* est, elle-aussi, γ-contractante pour la norme infinie, c'est à dire, que pour toutes applications v_1 et v_2 dans $\mathscr{C}(\mathscr{S},\mathbb{R})$,

$$\| L^*(v_2) - L^*(v_1) \|_\infty \leq \| v_2 - v_1 \|_\infty . \tag{III.12}$$

Pour vérifier ceci, remarquons que

$$\begin{aligned} \| L^*(v_2) - L^*(v_1) \|_\infty &= \max_{s\in\mathscr{S}} \left| \max_{a\in\mathscr{A}} \left\{ \bar{R}_1(s,a) + \gamma \mathbb{E}_{\Pi^*}\left[v_2(S_1)|S_0 = s; A_0 = a\right] \right\} - \max_{a\in\mathscr{A}} \left\{ \bar{R}_1(s,a) + \gamma \mathbb{E}_{\Pi^*}\left[v_1(S_1)|S_0 = s; A_0 = a\right] \right\} \right| \\ &\leq \max_{s\in\mathscr{S}} \max_{a\in\mathscr{A}} \left| \left\{ \bar{R}_1(s,a) + \gamma \mathbb{E}_{\Pi^*}\left[v_2(S_1)|S_0 = s; A_0 = a\right] \right\} - \left\{ \bar{R}_1(s,a) + \gamma \mathbb{E}_{\Pi^*}\left[v_1(S_1)|S_0 = s; A_0 = a\right] \right\} \right| \\ &= \max_{s\in\mathscr{S}} \max_{a\in\mathscr{A}} \gamma \left| \mathbb{E}_{\Pi^*}\left[v_2(S_1) - v_1(S_1)|S_0 = s; A_0 = a\right] \right| \\ &\leq \max_{s\in\mathscr{S}} \max_{a\in\mathscr{A}} \gamma \mathbb{E}_{\Pi^*}\left[\left| v_2(S_1) - v_1(S_1) \right| |S_0 = s; A_0 = a\right] \\ &\leq \max_{s\in\mathscr{S}} \max_{a\in\mathscr{A}} \gamma \mathbb{E}_{\Pi^*}\left[\| v_2 - v_1 \|_\infty |S_0 = s; A_0 = a\right] \\ &= \gamma \| v_2 - v_1 \|_\infty . \end{aligned}$$

On obtient donc (III.12), et comme $0 < \gamma < 1$, le théorème du point fixe de Banach implique à nouveau qu'il existe une unique solution v^* à l'équation de point fixe (III.11), et la convergence du schéma de puissance itérée vers cette solution. ■

On a alors le résultat fondamental suivant,

Théorème III.2.20. Caractérisation de la politique optimale (I)

Pour tout $\gamma \in [0,1)$ et toute politique stationnaire Π, Π est optimale si, et seulement si, v_{Π^*} coïncide avec l'unique solution v^* de l'équation de Bellman (III.11), autrement dit, pour tout $s \in \mathscr{S}$,

$$v_{\Pi^*}(s) = v^*(s) = L^* v^*(s) = \max_{a \in \mathscr{A}} \left\{ \bar{R}_1(s,a) + \gamma \mathbb{E}\left[v^*(S_1) | S_0 = s; A_0 = a \right] \right\}.$$

Démonstration. Partons d'un état $s \in \mathscr{S}$. Par la propriété de Markov, toute politique markovienne optimale consiste à prendre, à l'instant courant, l'action a^* qui optimise, en terme de récompense moyenne, la transition en un pas, c'est-à-dire,

(i) Prendre l'action a^*

(ii) Réaliser une transition de l'état s avec l'action a^* vers un nouvel état.

En effet, pour une telle politique Π^*, par la propriété de Markov on a alors

$$\begin{aligned}
v_{\Pi^*}(s) = \tilde{v}_{\Pi^*}(s,a^*) &= \mathbb{E}_{\Pi^*}\left[R_t \mid S_t = s; A_t = a^* \right] + \gamma \mathbb{E}_{\Pi^*}\left[G_{t+1} \mid S_t = s; A_t = a^* \right] \\
&= \mathbb{E}\left[R_{t+1}(S_t, A_t) \mid S_t = s; A_t = a^* \right] + \gamma \sum_{s' \in \mathscr{S}} \mathbb{E}_{\Pi^*}\left[G_{t+1} \mid S_{t+1} = s'; S_t = s; A_t = a \right] \mathbb{P}\left[S_{t+1} = s' | S_t = s; A_t = a^* \right] \\
&= \bar{R}_1(s,a^*) + \gamma \sum_{s' \in \mathscr{S}} \mathbb{E}_{\Pi^*}\left[G_{t+1} \mid S_{t+1} = s' \right] \mathbb{P}\left[S_{t+1} = s' | S_t = s; A_t = a^* \right] \\
&= \bar{R}_1(s,a^*) + \gamma \sum_{s' \in \mathscr{S}} v_{\Pi^*}(s') \mathbb{P}\left[S_{t+1} = s' | S_t = s; A_t = a^* \right] \\
&= \bar{R}_1(s,a^*) + \gamma \mathbb{E}\left[v_{\Pi^*}(S_{t+1}) | S_t = s; A_t = a^* \right] \\
&= \bar{R}_1(s,a^*) + \gamma \mathbb{E}\left[v_{\Pi^*}(S_1) | S_0 = s; A_0 = a^* \right].
\end{aligned}$$

et donc Π^* satisfait

$$v_{\Pi^*}(s) = \max_{a \in \mathscr{A}} \left\{ \bar{R}_1(s,a) + \gamma \mathbb{E}\left[v_{\Pi^*}(S_1) | S_0 = s; A_0 = a \right] \right\} = L^* v_{\Pi^*}(s), \quad s \in \mathscr{S}.$$

Par unicité de la solution à l'équation (III.11), on a donc $v_{\Pi^*}(s) = v^*(s)$ pour tout $s \in \mathscr{S}$. ■

Remarque III.2.21

La stratégie de construction de Π^* dans la preuve ci-dessus est souvent appelée **principe d'optimalité de Bellman** : "*An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.*"

Le Théorème III.2.20 montre donc que l'on peut obtenir la fonction de valeur optimale v_{Π^*} par un schéma itératif, donné dans la Proposition III.2.19.

Théorème III.2.22. Caractérisation de la politique optimale (II)

Pour tout $\gamma \in [0,1)$ et toute politique stationnaire Π, Π est optimale si, et seulement si, pour tout $s \in \mathscr{S}$, $\pi(.|s)$ admet pour support

$$\operatorname{argmax}_a \left\{ \bar{R}_1(s,a) + \gamma \mathbb{E}\left[v^*(S_1) | S_0 = s; A_0 = a \right] \right\},$$

où v^* est l'unique solution de (III.11).

Démonstration. Remarquons tout d'abord que la deuxième assertion du théorème peut se réécrire comme suit : pour tout s, $\pi(.|s)$ admet pour support l'ensemble des $a(s)$ qui réalisent le max

$$L^* v^*(s) = \max_{a\in\mathscr{A}}\left\{\bar{R}_1(s,a) + \gamma\mathbb{E}\left[v^*(S_1)|S_0 = s;\, A_0 = a\right]\right\},$$

et par conséquent,

$$\begin{aligned} L_\Pi v^*(s) &= \bar{R}_\Pi(s) + \gamma\mathbb{E}_\Pi\left[v^*(S_1)|S_0 = s\right] \\ &= \sum_{a\in\mathscr{A}}\left\{\bar{R}_1(s,a) + \gamma\mathbb{E}\left[v^*(S_1)|S_0 = s;\, A_0 = a\right]\right\}\pi(a|s) \\ &= \max_{a\in\mathscr{A}}\left\{\bar{R}_1(s,a) + \gamma\mathbb{E}\left[v^*(S_1)|S_0 = s;\, A_0 = a\right]\right\}\sum_{a\in\mathscr{A}}\pi(a|s) \\ &= \max_{a\in\mathscr{A}}\left\{\bar{R}_1(s,a) + \gamma\mathbb{E}\left[v^*(S_1)|S_0 = s;\, A_0 = a\right]\right\} = L^* v^*(s), \end{aligned}$$

ce qui revient à dire que

$$L_\Pi v^* = L^* v^*. \tag{III.13}$$

Nous en sommes ramenés à montrer que l'optimalité de Π est équivalente à (III.13).

"$\Longleftarrow$" Supposons tout d'abord que (III.13) est vérifiée. Comme on a aussi $L^* v^* = v^*$, on a donc que $L_\Pi v^* = v^*$, ce qui montre que v^* est solution de (III.4). Par unicité de ce point fixe, on a donc $v_\Pi = v^* = v_{\Pi^*}$, où la dernière égalité est précisément l'assertion du Théorème III.2.20. Π est donc optimale

"$\Longrightarrow$" Supposons maintenant que Π soit optimale, et donc que $v_\Pi = v^*$. Comme on a $L_\Pi v_\Pi = v_\Pi$, on a donc

$$L_\Pi v^* = v^* = L v^*,$$

ce qui conclut la preuve.

■

Une conséquence du résultat précédent est qu'il existe toujours (dans le cas $\mathscr{S}$ et $\mathscr{A}$ finis), une politique stationnaire *déterministe* qui soit optimale. Il suffit pour cela, pour tout état s, de prendre une action a avec proba 1, telle que $a(s)$ soit dans

$$\text{argmax}_a\left\{\bar{R}_1(s,a) + \gamma\mathbb{E}\left[v^*(S_1)|S_0 = s;\, A_0 = a\right]\right\},$$

où v^* est la fonction de valeur optimale. Nous venons donc de donner un premier algorithme pour calculer la fonction de valeur optimale v^*, appelé algorithme d'itération de valeurs (qui reprend donc le principe énoncé de façon préliminaire en Figure 1 :

Données: On se donne v^0. Soient $N^{\max} \in \mathbb{N}^*$ et $\varepsilon > 0$
On pose Res = 1 et $k = 0$;
Tant que Res $\geq \varepsilon$ *et* $k \leq N^{\max}$ **faire**
  On calcule v^{k+1} en résolvant le problème d'optimisation

$$\forall s \in \mathscr{S}, \qquad v^{k+1}(s) = \max_{a\in\mathscr{A}}\left\{\bar{R}_1(s,a) + \gamma\sum_{s'\in\mathscr{S}} v^k(s')\,T^a_{s\to s'}\right\}$$

  On pose Res $= \|v_{k+1} - v_k\|$ et $k \leftarrow k+1$
fin

Algorithme 3 : Algorithme d'itération de la valeur. Formulation stochastique.

L'algorithme 3 converge donc vers une fonction de valeur optimale. Il détermine aussi, pour tout $s \in \mathscr{S}$, l'action optimale $a^*(s)$ qui réalise le max. En pratique, l'algorithme a une complexité de $O(|\mathscr{S}|^2|\mathscr{A}|)$, puisqu'à chaque étape on doit faire une opération pour chaque triplet (s,s',a) dans $\mathscr{S}\times\mathscr{S}\times\mathscr{A}$, pour calculer la quantité $R_1 + \gamma v^k(s')T^a_{s\to s'}$. Mais on peut démontrer qu'il peut converge lentement, et d'autant plus lentement que les cardinaux de $\mathscr{S}$ et de $\mathscr{A}$ sont grands.

III.3 Itération de politiques

Définition III.3.1. Algorithme d'itération de politiques

Un algorithme d'**itération de politiques** a, informellement, les étapes suivantes :

(i) On part d'une règle de décision Π_0. Ensuite,

(ii) Pour $k = 0,1,2,\cdots$, connaissant la politique Π_k,

— *Evaluation de la fonction de valeur* : On détermine v_{Π_k} en résolvant le système

$$v_{\Pi_k} = L_{\Pi_k} v_{\Pi_k},$$

i.e., en posant

$$v_{\Pi_k} = (I - \gamma P_{\Pi_k})^{-1} \bar{R}_{\Pi_k}.$$

— *Mise à jour de la politique* : On détermine Π_{k+1} en résolvant l'équation

$$L_{\Pi_{k+1}} v_{\Pi_k} = L^* v_{\Pi_k}$$

de façon 'gloutonne', c'est à dire, on pose pour tout $s \in \mathscr{S}$, $\pi_{k+1}(a(s)|s) = 1$, pour un $a(s)$ dans

$$\text{argmax}_a \left\{ \bar{R}_1(s,a) + \gamma \mathbb{E}\left[v_{\Pi_k}(S_1) | S_0 = s; A_0 = a \right] \right\}.$$

(iii) On s'arrête quand $v_{\Pi_{k+1}}$ et v_{Π_k} sont suffisamment proches.

Cela donne l'algorithme suivant,

Données: On se donne Π et v. On fixe $N^{\max} \in \mathbb{N}^*$ et $\varepsilon > 0$.
Faire $1 \leftarrow$ Res et $k \leftarrow 0$.
Tant que Res $\geq \varepsilon$ *et* $k \leq N^{\max}$ **faire**
Calculer

$$v' = (I - \gamma P_\Pi)^{-1} \bar{R}_\Pi$$

Poser

$$\text{Res} = \| v' - v \|$$

Poser, pour tout $s \in \mathscr{S}$, $\pi(a(s)|s) = 1$ pour un $a(s)$ dans

$$\text{argmax}_a \left\{ \bar{R}_1(s,a) + \gamma \mathbb{E}\left[v'(S_1) | S_0 = s; A_0 = a \right] \right\}.$$

Faire $v \leftarrow v'$ et $k \leftarrow k+1$
fin

Algorithme 4 : Algorithme d'itération de la politique.

Dans l'étape d'évaluation de la fonction de valeur, on peut préférer recourir au schéma itératif de l'algorithme 2 pour calculer v_{Π_k}. On remplace alors l'algorithme précédent par le suivant.

Données: On se donne Π, v et v'. On fixe $N^{\max} \in \mathbb{N}^*$ et $\varepsilon > 0$.
Poser Res = 1 et $k = 0$.
Tant que Res $\geq \varepsilon$ *et* $k \leq N^{\max}$ **faire**
  Poser Res' = 1 ;
  Tant que Res' $\geq \varepsilon$ **faire**

$$v' \leftarrow v$$

  Pour tout $s \in \mathscr{S}$ Calculer

$$v'(s) = \sum_{a \in \mathscr{A}} \left\{ \bar{R}_1(s,a) + \gamma \sum_{s' \in \mathscr{S}} v(s') T^a_{s \to s'} \right\} \pi_k(a|s)$$

  Poser Res' $= \| v' - v \|$
  fin
  Poser

$$\text{Res} = \| v' - v \|$$

  Poser, pour tout $s \in \mathscr{S}$, $\pi(a(s)|s) = 1$ pour un $a(s)$ dans

$$\operatorname{argmax}_a \left\{ \bar{R}_1(s,a) + \gamma \mathbb{E}\left[v'(S_1) | S_0 = s;\, A_0 = a \right] \right\}.$$

  Faire $v \leftarrow v'$ et $k \leftarrow k+1$
fin

Algorithme 5 : Algorithme d'itération de la politique avec estimation de la fonction de valeur.

Proposition III.3.2. Monotonie de l'algorithme 4

L'algorithme 4 (ou 5) génère des fonctions de valeurs croissantes point par point :

$$\text{Pour tout } k \in \mathbb{N}, \text{ pour tout } s \in \mathscr{S}, \quad v_{\Pi_{k+1}}(s) \geq v_{\Pi_k}(s).$$

Démonstration. La monotonie découle du fait que pour tout k,

$$\begin{aligned} v_{\Pi_{k+1}} - v_{\Pi_k} &= (I - \gamma P_{\Pi_{k+1}})^{-1} \bar{R}_{\Pi_{k+1}} - v_{\Pi_k} \\ &= (I - \gamma P_{\Pi_{k+1}})^{-1} \left(\bar{R}_{\Pi_{k+1}} - (I - \gamma P_{\Pi_{k+1}}) v_{\Pi_k} \right) \\ &= (I - \gamma P_{\Pi_{k+1}})^{-1} \left(\bar{R}_{\Pi_{k+1}} + \gamma P_{\Pi_{k+1}} v_{\Pi_k} - v_{\Pi_k} \right) \\ &= (I - \gamma P_{\Pi_{k+1}})^{-1} \left(L_{\Pi_{k+1}} v_{\Pi_k} - v_{\Pi_k} \right) \\ &= (I - \gamma P_{\Pi_{k+1}})^{-1} \left(L^* v_{\Pi_k} - L_{\Pi_k} v_{\Pi_k} \right). \end{aligned} \tag{III.14}$$

Or, d'une part, par définition de L_{Π_k} on a pour tout $s \in \mathscr{S}$,

$$\begin{aligned} L_{\Pi_k} v_{\Pi_k}(s) &= \bar{R}_{\Pi_k}(s) + \gamma \mathbb{E}_\Pi \left[v_{\Pi_k}(S_1) | S_0 = s \right] \\ &= \left\{ \bar{R}_1(s) + \gamma \mathbb{E}_\Pi \left[v_{\Pi_k}(S_1) | S_0 = s;\, A_0 = a \right] \right\} \pi_k(a|s) \\ &\leq \max_{a \in \mathscr{A}} \left\{ \bar{R}_1(s) + \gamma \mathbb{E}_\Pi \left[v_{\Pi_k}(S_1) | S_0 = s;\, A_0 = a \right] \right\} \sum_{a \in \mathscr{A}} \pi_k(a|s) \\ &= \max_{a \in \mathscr{A}} \left\{ \bar{R}_1(s) + \gamma \mathbb{E}_\Pi \left[v_{\Pi_k}(S_1) | S_0 = s;\, A_0 = a \right] \right\} \\ &= L^* v_{\Pi_k}(s), \end{aligned}$$

et donc le vecteur du membre de droite de (III.14) a des coordonnées positives ou nulles. D'autre part, la matrice

$$(I - \gamma P_{\Pi_{k+1}})^{-1} = \sum_{i=0}^{\infty} \gamma^i (P_{\Pi_{k+1}})^i$$

est définie positive, ce qui montre que $v_{\Pi_{k+1}} - v_{\Pi_k}$ est à coordonnées positives ou nulles, d'où la monotonie. ■

L'algorithme 4 a une complexité de $O(|\mathscr{S}|^2|\mathscr{A}|) + O(|\mathscr{S}|^3)$, puisqu'en plus du choix de a pour tout s, à chaque étape on inverse une matrice carrée de taille $|\mathscr{S}|$. On a le résultat suivant, voir [4],

Théorème III.3.3. Convergence de l'algorithme 4

L'algorithme 4 (ou 5) converge vers la fonction de valeur optimale v^* en au plus $O\left(\frac{|\mathscr{S}||\mathscr{A}|}{1-\gamma}\log\frac{1}{1-\gamma}\right)$ itérations, au bout desquelles on a même $v_{\Pi_{k+1}} = v_{\Pi_k} = v^*$.

Démonstration. La preuve de l'arrêt au bout d'un nombre fini de pas repose sur le fait que l'algorithme 5 peut se réécrire avec le critère d'arrêt suivant.

Données: On se donne Π une politique déterministe, v et v'. On fixe $N^{\max} \in \mathbb{N}^*$ et $\varepsilon > 0$.
On pose Res = 1 et Policy-stable = Faux.
Tant que Policy-stable=Faux **faire**
  Poser Res' = 1 ;
  Tant que Res' $\geq \varepsilon$ **faire**

$$v' \leftarrow v$$

    Pour tout $s \in \mathscr{S}$ calculer

$$v'(s) = \sum_{a\in\mathscr{A}} \left\{ \bar{R}_1(s,a) + \gamma \sum_{s'\in\mathscr{S}} v(s') T^a_{s\to s'} \right\} \pi_k(a|s).$$

    Poser Res' = $\|v' - v\|$
  fin

$$\text{Policy-stable} \leftarrow \text{Vrai}$$

  Pour $s \in \mathscr{S}$ **faire**
    Poser ancienne action $\leftarrow a(s)$ t.q. $\Pi(a(s)|s) = 1$.
    Poser $\pi(a(s)|s) = 1$ pour un $a(s)$ dans $\text{argmax}_a \left\{ \bar{R}_1(s,a) + \gamma \mathbb{E}\left[v_\Pi(S_1)|S_0 = s; A_0 = a\right] \right\}$.
    if ancienne action $\neq a(s)$ **then**
      Policy-stable $\leftarrow$ Faux.
    end
  fin
fin
Renvoyer Π et v'.

Algorithme 6 : Algorithme d'itération de la politique avec estimation de la fonction de valeur et critère d'arrêt sur la politique.

Dans l'Algorithme 6, le critère d'arrêt porte sur l'action déterministe prise en tout état. Comme les ensembles $\mathscr{S}$ et $\mathscr{A}$ sont finis, l'algorithme termine en un nombre fini d'itérations. On a alors $\Pi_{k+1} = \Pi_k$ et donc $v_{\Pi_{k+1}} = v_{\Pi_k}$. D'autre part, l'algorithme converge vers la solution optimale, puisque pour tout k, si $v_{\Pi_{k+1}} = v_{\Pi_k}$ on a

$$v_{\Pi_k} = v_{\Pi_{k+1}} = L_{\Pi_{k+1}} v_{\Pi_{k+1}} = L_{\Pi_{k+1}} v_{\Pi_k} = L^* v_{\Pi_k},$$

et donc $v_{\Pi_{k+1}} = v_{\Pi_k} = v^*$ par unicité de la solution à (III.11). L'ordre du nombre d'opérations, est admis. ■

III.4 Problèmes sans modèles

Nous considérons maintenant que les $T^a_{s\to s'}$ et par là-même, les transitions de la chaîne de Markov sous-jacente, ne sont pas connues. Ceci peut à la fois être une contrainte du modèle (on ne connaît concrètement pas les transitions), ou un choix d'approche, puisque la connaissance des transitions et les calculs liés aux algorithmes 3 ou 4 sont trop lourds.

Dans ce contexte, dit sans modèle (*model free*), si l'on se base sur l'architecture de l'algorithme 4, on doit donc faire à chaque étape, deux opérations :

(i) Estimer la fonction de valeur pour un MDP dont les transitions sont inconnues, et donc implémenter une version "approchée", si c'est possible, de l'algorithme 3 (étape de *prédiction*) ;

(ii) Améliorer la politique (étape de *contrôle*).

III.4.1 Prédiction par Monte-Carlo

Pour l'étape de prédiction, les méthodes de Monte-Carlo consistent en une estimation de l'espérance

$$v_\Pi(s) = \mathbb{E}_\Pi\left[G_0 | S_0 = s\right], s \in \mathscr{S},$$

en utilisant des moyennes empiriques. Pour ceci, en supposant que la politique π est stationnaire, on fixe un horizon T, et on simule un n-échantillon de la fonction objectif à horizon T, i.e.

$$G(s) := G(s) = \sum_{k=0}^{T\wedge\tau} \gamma^k R_{1+k}, \text{ partant de } s \in \mathscr{S},$$

où $\tau = \inf\{t \in \mathbb{N} : S_t \in \mathscr{S}'\}$, le temps d'atteinte d'un premier état absorbant par la chaîne $(S_t)_{t\in\mathbb{N}}$ (qui est donc un temps d'arrêt). $G(s)$ est donc une variable aléatoire qui dépend de Π et des transitions (supposées inconnues) $T^a_{s\to s'}$. En d'autres termes, pour tout $i = 1,...,n$, on tire une réalisation $G_i(s)$ de la variable aléatoire précédente, appelée v.a. *cible*. Alors, en rappelant que l'on a pour tout s, $v_\Pi(s) = \mathbb{E}_\Pi[G(s)]$ on estime la fonction de valeur par

$$\overline{v^n_\Pi}(s) := \frac{1}{n}\sum_{i=1}^{n} G_i(s), \quad s \in \mathscr{S},$$

dont la loi forte des grands nombres nous apprend qu'il tend p.s. vers $\mathbb{E}_\Pi[G(s)]$. De la même manière, on peut définir l'estimation Monte-Carlo de la fonction de valeur état/action, par

$$\overline{\tilde{v}^n_\Pi}(s,a) := \frac{1}{n}\sum_{i=1}^{n} G_i(s,a), \quad s \in \mathscr{S}, a \in \mathscr{A}, \tag{III.15}$$

où les $G_i(s,a)$, $i = 1,...,n$, forment i réalisations de la fonction objectif partant de l'état s et de l'action a. Remarquons que l'on a, pour tout $s \in \mathscr{S}$ et tout $n \in \mathbb{N}$,

$$\begin{aligned}
\overline{v^{n+1}_\Pi}(s) &= \frac{n}{n+1}\overline{v^n_\Pi}(s) + \frac{1}{n+1}G^T_{n+1} \\
&= \frac{n+1}{n+1}\overline{v^n_\Pi}(s) - \frac{1}{n+1}\overline{v^n_\Pi}(s) + \frac{1}{n+1}G^T_{n+1} \\
&= \overline{v^n_\Pi}(s) + \frac{1}{n+1}\left(G^T_{n+1} - \overline{v^n_\Pi}(s)\right) \\
\iff \text{Nouvelle estimation} &= \text{Ancienne estimation} + \text{Taille de pas}\,(\text{cible - Ancienne estimation}),
\end{aligned} \tag{III.16}$$

et de même pour la fonction de valeur état/action.

Ces estimations de la fonction de valeur sont biaisées, et optimale au sens des moindres carrés. En outre, l'intérêt de cette estimation est de pouvoir estimer la fonction de valeur en tout point :

- *Sans* tenir compte des estimations faites dans les points de l'espace d'état ;
- *Sans* utiliser la propriété de Markov de $(S_t)_{t\in\mathbb{N}}$.

III.4.2 Prédiction par Différences temporelles

On peut reproduire le schéma de l'approche Monte-Carlo de (III.16), en se passant de simuler sur un long intervalle de temps, mais en mettant à jour la fonction de valeur à chaque pas de temps, au gré de la chaîne de Markov $(S_t)_{t\in\mathbb{N}}$. On met alors à jour la fonction de valeur suivant le schéma suivant : pour un certain incrément α,

$$\widehat{v_\Pi}(S_t) \longleftarrow \widehat{v_\Pi}(S_t) + \alpha\left(G_t - \widehat{v_\Pi}(S_t)\right),$$

en rappelant (III.5). On a formellement, la définition suivante,

Définition III.4.1. Schéma Itératif par différence temporelle

Soit Π une politique stationnaire. Soit $\alpha > 0$. L'estimation de la fonction de valeur v_Π à la t-ième itération, est donnée par le schéma itératif $((\widehat{v_\Pi}(S_t), S_t))_{t\in\mathbb{N}}$ suivant, pour une valeur initiale (v,s),

$$(\widehat{v_\Pi}(S_0), S_0) \longleftarrow (v(s), s);$$
$$(\widehat{v_\Pi}(S_{t+1}), S_{t+1}) \longleftarrow \left(\widehat{v_\Pi}(S_t) + \alpha\left(R_{t+1} + \gamma\widehat{v_\Pi}(S_{t+1})\right) - \widehat{v_\Pi}(S_t), S_{t+1}\right), t \in \mathbb{N}.$$

Ceci donne donc l'algorithme suivant, appelé TD(0),

Données: On se donne Π, $N^{\max}$, α, v et s
On pose $V = v, S = s, N = 0$;
Tant que *On n'atteint pas un état absorbant et* $N \leq N^{\max}$ **faire**
  $A \longleftarrow a$ telle que $\pi(a|s) = 1$.
  Prendre l'action A partant de S et observer R et S'.
  $V(S) \longleftarrow V(S) + \alpha\left(R + \gamma V(S') - V(S)\right)$.
  $S \longleftarrow S'$
  $N \longleftarrow N + 1$.
fin

Algorithme 7 : Algorithme TD(0).

L'algorithme TD(0) fournit une estimation biaisée de la fonction de valeur sous la politique Π (comme l'estimation par Monte-Carlo), mais ayant une plus petite variance. En outre, l'estimation par TD(0) est l'Estimateur du Maximum de Vraissemblance, si l'on suppose que le processus des états $(S_t)_{t\in\mathbb{N}}$ est markovien (même si ses transitions sont inconnues...) et ergodique. Alors, par la théorie de l'approximation stochastique on peut montrer que sous de bonnes conditions, la solution V renvoyée par l'algorithme 7 converge vers la solution du système

$$\dot{v} = r + (\gamma P_\Pi - I)v,$$

et par là-même, vers l'unique fonction de valeur v_Π.

Les Figures III.4.2, III.4.2 et III.4.2 (dues à [3]), montrent les différents schémas d'exploration des espaces des états/actions par :

- La programmation dynamique et les MDP (*DP Backup* - Figure III.4.2), où l'on fait la moyenne sur tous les états suivants ;
- Une méthode Monte-Carlo, où l'on estime la cible en explorant tout une trajectoire jusqu'à un éventuel point absorbant (*MC Backup* - Figure III.4.2) ;
- Une méthode de type Différence Temporelle, où l'on explore et estime la cible en un pas de temps (*TD Backup* - Figure III.4.2).

Exemple III.4.2 Marche aléatoire symétrique

Considérons l'exemple, du à [5], d'une marche aléatoire symétrique vers 2 états absorbants opposés, de récompenses respectives 0 et 1, sans action. La fonction de valeur est donc simplement la probabilité de se retrouver dans l'état absorbant de récompense 1, en partant des différents états initiaux A, B, C, D et E. Il est facile de vérifier que la fonction de valeur (unique, car toutes les politiques sont équivalentes en l'absence d'action...) sont respectivement données par

$$V(A) = \frac{1}{6}, V(B) = \frac{2}{6}, V(C) = \frac{3}{6}, V(D) = \frac{4}{6}, V(E) = \frac{5}{6}.$$

La Figure III.4.2 montre la convergence, et sa vitesse, vers la vraie fonction de valeur, par l'algorithme MC et par l'algorithme TD(0) pour différentes valeurs de α.

85

DP backup

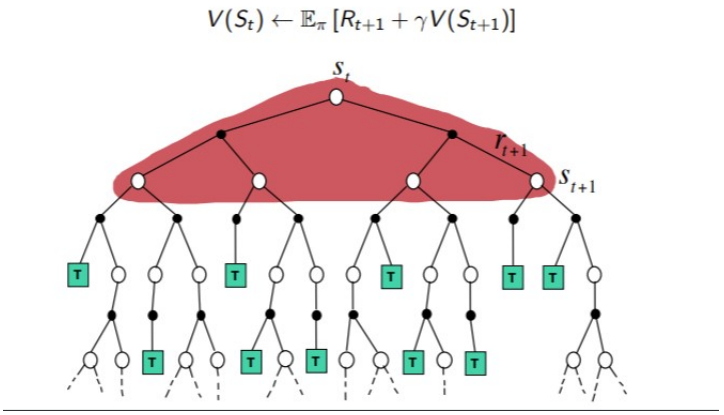


FIGURE III.1 : *Exploration de l'espace des états/actions par un MDP*

86

MC backup

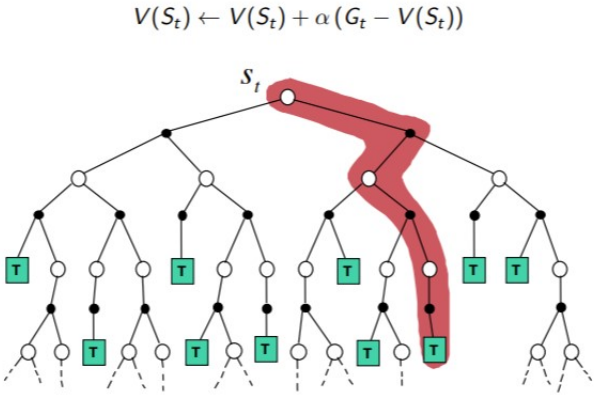


FIGURE III.2 : *Exploration de l'espace des états/actions par une méthode Monte-Carlo*

On peut améliorer la vitesse de convergence vers cette solution, en proposant un "mix" des deux approches permettant d'explorer l'espace d'états de manière plus exhaustive, en prenant comme v.a. cible, la v.a. G^n pour un entier fixé n. Ceci mène à l'algorithme suivant, appelé TD(n).

87

TD backup

$$V(S_t) \leftarrow V(S_t) + \alpha\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)$$

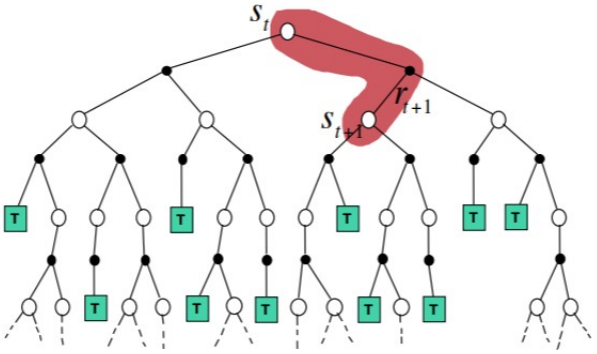
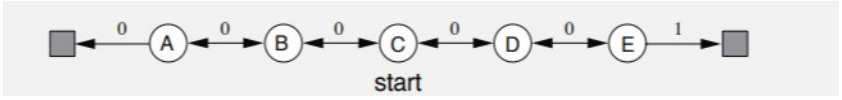


FIGURE III.3 : *Exploration de l'espace des états/actions par une méthode de différence temporelle*

89

Comparisons MC vs TD on random walk reward process



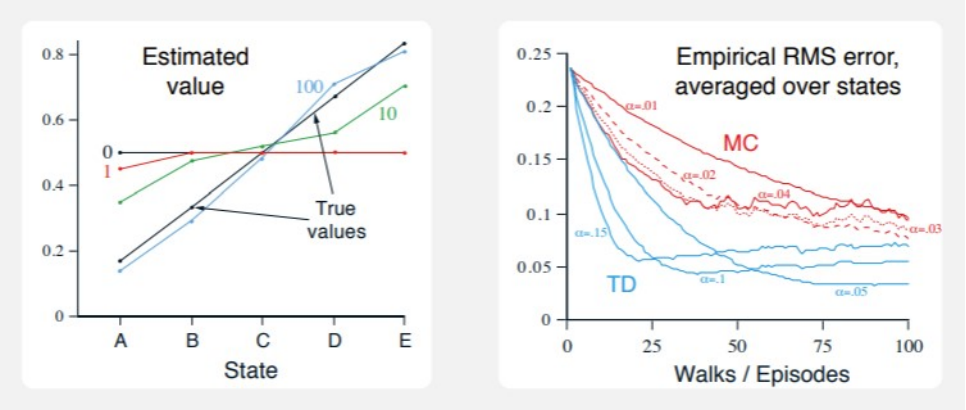


FIGURE III.4 : *Convergence vers la fonction de valeur dans le cas d'une marche aléatoire symétrique sans actions et avec états absorbants*

Données: On se donne Π, $N^{\max}$, α, v et s
On pose $V = v, S = s, N = 0$;
Tant que *On n'atteint pas un état absorbant et* $N \leq N^{\max}$ **faire**
  $A \longleftarrow a$ telle que $\pi(a|s) = 1$.
  Prendre l'action A partant de S et observer R et S'.
  $V(S) \longleftarrow V(S) + \alpha\left(R_1 + \gamma R_2 + \cdots + \gamma^n R_{n+1} - V(S)\right)$.
  $S \longleftarrow S'$
  $N \longleftarrow N + 1$.
fin

Algorithme 8 : Algorithme TD(n).

III.4.3 Contrôle par Monte-Carlo

Comme évoqué plus haut, on peut également chercher à optimiser la politique à la volée, tout en estimant la fonction de valeur correspondante, au fur et à mesure de l'exploration de l'espace d'état et de l'espace des actions. C'est l'étape dite de *contrôle*. Il y a deux types de contrôle principaux :

- Dans le cas *online*, la politique que l'on est en train d'optimiser est aussi celle qui permet l'exploration (car les décisions prises conditionnent la visite des différents états).
- Dans le cas *offline*, une politique spécifique est utilisée pour l'exploration, et on optimise la politique grâce à cette exploration.

L'idée du contrôle par Monte-Carlo est de proposer un contrôle *online*. dont l'idée informelle est la suivante : À chaque étape k de l'algorithme, on part d'un couple état/action (s,a) et :

(i) On fait l'estimation Monte-Carlo de la fonction de valeur état/action par la moyenne des récompenses immédiatement obtenues dans les différents couples état/action rencontrés.

(ii) On améliore la politique en prenant l'action a qui maximise la fonction de valeur état/action, pour tout état $s \in \mathscr{S}$.

Ceci donne l'algorithme de contrôle par Monte-Carlo suivant, où il est supposé implicitement que $\gamma = 1$, ce qui est particulièrement pertinent dans le contexte où T n'est pas trop grand et/ou de points absorbants :

Données: On se donne Π, $N^{\max}$ et ϵ.
On pose pour tous $s \in \mathscr{S}$, $a \in \mathscr{A}$,
$\tilde{v}(s,a) \leftarrow v(s,a)$,
Returns$(s,a) \leftarrow \emptyset$.
Tant que $N \leq N^{\max}$ **faire**
  Générer une trajectoire jusqu'à $T \wedge \tau$;
  Pour *Paires* (s,a) *rencontrées pendant la trajectoire* **faire**
    $G \leftarrow$ Récompense immédiatement obtenue en quittant(s,a)
    Returns$(s,a) \leftarrow$ (Returns(s,a),G)
    $\tilde{v}(s,a) \leftarrow$ moyenne (Returns(s,a)).
  fin
  Pour *Etats s rencontrés pendant la trajectoire* **faire**
    $A^* \longleftarrow a$ tel que $a \in \operatorname{argmax}_a \tilde{v}(s,a)$
    Poser $\pi(A^*|s) = 1$.
  fin
  $N \longleftarrow N + 1$.
fin

Algorithme 9 : Algorithme *online* MC.

On rencontre alors une problématique d'exploration/exploitation : il faut visiter suffisamment d'actions, dans chaque état, pour que cette politique puisse converger vers la politique optimale sans "négliger" des actions. On peut alors remplacer l'étape (iii) de la procédure ci-dessus par une prise d'action de type ϵ-*greedy* : on on prend une action de l'Argmax ci-dessus avec probability ϵ, ou alors on repart de n'importe quelle autre action uniformément au hasard, avec probabilité $1 - \varepsilon$, où ϵ est un paramètre à éventuellement optimiser.

Données: On se donne Π, $N^{\max}$ et ϵ.
On pose pour tous $s \in \mathscr{S}$, $a \in \mathscr{A}$,
$\tilde{v}(s,a) \leftarrow v(s,a)$,
Returns$(s,a) \leftarrow \emptyset$.
Tant que $N \leq N^{\max}$ **faire**
  Générer une trajectoire jusqu'à $T \wedge \tau$;
  Pour *Paires* (s,a) *rencontrées pendant la trajectoire* **faire**
    $G \leftarrow$ Récompense immédiatement obtenue en quittant (s,a)
    Returns$(s,a) \leftarrow$ (Returns(s,a),G)
    $\tilde{v}(s,a) \leftarrow$ moyenne (Returns(s,a)).
  fin
  Pour *Etats s rencontrés pendant la trajectoire* **faire**
    $A^* \longleftarrow a$ tel que $a \in \operatorname{argmax}_a \tilde{v}(s,a)$
    Pour $a \in \mathscr{A}(s)$ **faire**
      Poser $\pi(a|s) = \begin{cases} 1-\epsilon+\frac{\epsilon}{|\mathscr{A}(s)|} & \text{si } a = A^*; \\ \frac{\epsilon}{|\mathscr{A}(s)|} & \text{si } a \neq A^*. \end{cases}$.
    fin
  fin
  $N \longleftarrow N+1$.
fin

Algorithme 10 : Algorithme *online* MC ϵ-*greedy*.

Il n'existe pas, en toute généralité, de résultat montrant la convergence des algorithmes 9 et . En revanche, on peut montrer qu'ils convergent si ϵ décroît dans le temps suffisamment vite et par ailleurs, que s'ils convergent, c'est vers une politique réalisant la fonction de valeur optimale. Pour le montrer, remarquons tout d'abord le résultat suivant.

Proposition III.4.3.

Pour tout N dans les algorithmes 9 et III.4.3, en notant Π la politique dont on part en début de l'étape n et Π' celle que l'on a construit à la fin de l'étape n, on a pour tout $s \in \mathscr{S}$,

$$\mathbb{E}_{\Pi'}[\tilde{v}_\Pi(s,A)] \geq v_\Pi(s).$$

Démonstration. Pour tout $\epsilon \in [0,1)$ (le cas $\epsilon = 0$ représenant celui de l'algorithme 9), on a pour tout $s \in \mathscr{S}$,

$$\begin{aligned}
\mathbb{E}_{\Pi'}[\tilde{v}_\Pi(s,A)] &= \sum_{a\in\mathscr{A}(s)} \tilde{v}_\Pi(s,a)\pi'(a|s) \\
&= \frac{\epsilon}{|\mathscr{A}(s)|} \sum_{a\in\mathscr{A}(s)\setminus\{A^*\}} \tilde{v}_\Pi(s,a) + \frac{\epsilon}{|\mathscr{A}(s)|}\tilde{v}_\Pi(s,A^*) + (1-\epsilon)\tilde{v}_\Pi(s,A^*) \\
&= \frac{\epsilon}{|\mathscr{A}(s)|} \sum_{a\in\mathscr{A}(s)} \tilde{v}_\Pi(s,a) + (1-\epsilon)\max_a \tilde{v}_\Pi(s,a) \\
&= \frac{\epsilon}{|\mathscr{A}(s)|} \sum_{a\in\mathscr{A}(s)} \tilde{v}_\Pi(s,a) + (1-\epsilon)\left(\sum_{a\in\mathscr{A}(s)} \frac{\pi(a|s)-\frac{\epsilon}{|\mathscr{A}(s)|}}{1-\epsilon}\right)\max_a \tilde{v}_\Pi(s,a) \\
&\geq \frac{\epsilon}{|\mathscr{A}(s)|} \sum_{a\in\mathscr{A}(s)} \tilde{v}_\Pi(s,a) + (1-\epsilon)\sum_{a\in\mathscr{A}(s)} \frac{\pi(a|s)-\frac{\epsilon}{|\mathscr{A}(s)|}}{1-\epsilon}\tilde{v}_\Pi(s,a) \\
&= \sum_{a\in\mathscr{A}(s)} \tilde{v}_\Pi(s,a)\pi(a|s) \\
&= v_\Pi(s),
\end{aligned}$$

où l'on utilise que la deuxième somme du terme de droite de la quatrième égalité, vaut 1. ■

Ceci implique que l'algorithme III.4.3 entraîne une amélioration continuelle de la version de valeur.

Corollaire III.4.4.

Sous les hypothèses de la Proposition III.4.3, on a pour tout $s \in \mathscr{S}$, $v_\Pi(s) \geq v_{\Pi'}(s)$.

Démonstration. On a pour tout $s \in \mathscr{S}$, en notant "$A_0 = \pi'(.|s)$" pour noter l'événement "A_0 est tiré au sort suivant la loi $\pi'(.|s)$'

$$\begin{aligned}
v_\Pi(s) &\leq \sum_{a \in \mathscr{A}(s)} \tilde{v}_\Pi(s,a)\pi'(a|s) \\
&= \sum_{a \in \mathscr{A}(s)} \mathbb{E}_\Pi\left[R_1 + \gamma v_\Pi(S_1) | S_0 = s, A_0 = a\right] \pi'(a|s) \\
&= \mathbb{E}_{\Pi'}\left[R_1 + \gamma v_\Pi(S_1) | S_0 = s\right] \\
&\leq \mathbb{E}_{\Pi'}\left[R_1 + \gamma \mathbb{E}_{\Pi'}\left[\tilde{v}_\Pi(S_1, A)\right] | S_0 = s\right] \\
&= \mathbb{E}_{\Pi'}\left[R_1 + \gamma \mathbb{E}_{\Pi'}\left[R_2 + \gamma v_\Pi(S_2)\right] | S_0 = s\right] \\
&= \mathbb{E}_{\Pi'}\left[R_1 + \gamma R_2 + \gamma^2 v_\Pi(S_2) | S_0 = s\right] \\
&\leq \mathbb{E}_{\Pi'}\left[R_1 + \gamma R_2 + \gamma^2 v_\Pi(S_2) + \gamma^3 v_\Pi(S_3) | S_0 = s\right] \\
&... \\
&\leq \mathbb{E}_{\Pi'}\left[R_1 + \gamma R_2 + \gamma^2 v_\Pi(S_2) + \gamma^3 v_\Pi(S_3) + \gamma^4 v_\Pi(S_4) + ... | S_0 = s\right] \\
&= \mathbb{E}_{\Pi'}\left[G_0 | S_0 = s\right] \\
&= v_\Pi(s).
\end{aligned}$$

■

Par conséquent, si l'algorithme converge, il converge en croissant (pour la fonction de valeur) vers une politique optimale.

III.4.4 Contrôle par différences temporelles

Nous terminons par deux algorithmes de contrôle fondés sur la technique des différences temporelles. Le premier est un algorithme *online*, où l'on explore à nouveau au fur et à mesure avec la politique en cours d'optimisation. Il est appelé SARSA pour

$$\text{(state,action)} \longrightarrow \text{(reward,state)} \longrightarrow \text{action}.$$

On procède comme suit : tant que l'on atteint pas l'instant terminal $T \wedge \tau$ d'une trajectoire, on parcours les couples (état/action) (s,a) en observant la récompense et le nouvel état s', en prenant une nouvelle action a' par un critère (par exemple ϵ-greedy) dépendant de $\tilde{v}_\Pi(s,a)$, et en mettant à jour $\tilde{v}_\Pi(s,a)$ par différence temporelle. Il se détaille comme suit :

Données: On se donne $\tilde{V}$ sur $\mathscr{S} \times \mathscr{A}$ tel que $\tilde{V}(s,a) = 0$ pour tout a et tout $s \in \mathscr{S}'$.
On se donne $N^{\max}$ et α.
Tant que $N \leq N^{\max}$ **faire**
  Fixer $S \in \mathscr{S}$ et $A \in \mathscr{A}(S)$ tiré au sort par ϵ-greedy en fonction de $\tilde{V}$.
  Tant que $t \leq T \wedge T$ *et* $S \in \mathscr{S} \setminus \mathscr{S}'$ **faire**
    Prendre l'action A et observer R et le nouvel état S'
    Choisir $A' \in \mathscr{A}(S')$ tiré au sort par ϵ-greedy en fonction de $\tilde{V}$
    Mettre à jour $\tilde{V}(S,A) \leftarrow \tilde{V}(S,A) + \alpha\left[R + \gamma \tilde{V}(S',A') - \tilde{V}(S,A)\right]$
    $S \leftarrow S'$
    $A \leftarrow A'$.
  fin
  $N \longleftarrow N+1$.
fin
.

Algorithme 11 : Algorithme SARSA

On peut montrer que, sous de bonnes conditions, SARSA converge vers la politique optimale. On considère finalement la politique suivante, qui est une politique de type *offline*, où

- On explore en appliquant ϵ-greedy en fonction de $\tilde{v}_\Pi$;
- On met à jour avec une cible différente, qui est simplement greedy ($\epsilon = 0$) en fonction de $\tilde{v}_\Pi$ (et donc, on ne choisit pas l'action donnée par ϵ-greedy dans la cible, mais une autre).

Ceci donne la variante suivante, appelée algorithme de Q-learning,

Données: On se donne $\tilde{V}$ sur $\mathscr{S} \times \mathscr{A}$ tel que $\tilde{V}(s,a) = 0$ pour tout a et tout $s \in \mathscr{S}'$.
On se donne $N^{\max}$ et α.
Tant que $N \leq N^{\max}$ **faire**
  Fixer $S \in \mathscr{S}$. **Tant que** $t \leq T \wedge T$ *et* $S \in \mathscr{S} \setminus \mathscr{S}'$ **faire**
    Prendre l'action A à partir de S en fonction de $\tilde{V}$ par ϵ-greedy
    Observer R, S'.
    Mettre à jour $\tilde{V}(S,A) \leftarrow \tilde{V}(S,A) + \alpha\left[R + \gamma \max_{a \in \mathscr{A}(S')} \tilde{V}(S',a) - \tilde{V}(S,A)\right]$
    $S \leftarrow S'$.
  fin
  $N \longleftarrow N+1$.
fin
.

Algorithme 12 : Algorithme Q-learning

III.5 Exercices du chapitre

III.1 Dans l'exemple du Green Robot (Exemple III.2.2),

(i) Construire une politique markovienne, mais non stationnaire.
(ii) Construire une politique markovienne et stationnaire.
(iii) Construire une politique non stationnaire et déterministe.
(iv) Construire une politique stationnaire et déterministe.

III.2 Dans l'exemple du Green Robot (Exemple III.2.2), on suppose que la politique est la suivante :

(i) Si on est dans l'état *low*, on fait un tirage au sort indépendant de tout le reste et :
  - Avec probabilité 1/2, le robot va se charger ;
  - Avec probabilité 1/4, le robot cherche les déchets ;
  - Avec probabilité 1/4, le robot attend.

(ii) Si on est dans l'état *high*, on fait un tirage au sort indépendant de tout le reste et :
  - Avec probabilité 3/4, le robot cherche les déchets.
  - Avec probabilité 1/4, le robot attend.

(i) Montrer que la politique proposée est markovienne, stationnaire et déterministe.
(ii) Donner alors les probabilités de transition de la chaîne de Markov $(S_t)_{t \in \mathbb{N}}$.
(iii) Calculer la fonction de valeurs en tout point.
(iv) Calculer la fonction de valeurs état/action en tout point.

III.3 Démontrer la Proposition III.2.7

III.4 Dans l'exemple du Green Robot (Exemple III.2.2),

(i) Pour une politique markovienne stationnaire Π donnée, exprimer la fonction de valeur v_Π comme solution d'un système linéaire dont on précisera les inconnues et les équations.
(ii) Poser (sous forme de système) l'équation de Bellman satisfaite par la fonction de valeur optimale v^*.

III.5 Dans l'exemple du dilemme de l'étudiant (Exemple III.2.16),

(i) Modéliser les actions et transitions de la Figure 2 par un processus de décision markovien.

(ii) Pour une politique markovienne stationnaire Π donnée, exprimer la fonction de valeur v_Π comme solution d'un système linéaire dont on précisera les inconnues et les équations.

III.6 Dans l'exemple du dilemme de l'étudiant (Exemple III.2.16),

(i) On considère la politique consistant à travailler tout le temps. Calculer sa fonction de valeur par une inversion matricielle.

(ii) Retrouver ce résultat par un algorithme de type puissance itérée.

(iii) Mêmes questions pour la politique déterministe suivante :

$$\pi(R|1) = 1; \pi(W|2) = 1; \pi(W|3) = 1; \pi(R|4) = 1.$$

(iv) Mêmes questions pour la politique suivante, non déterministe :

$$\begin{cases} \pi(R|1) & = 0.8; \pi(W|1) = 0.2; \\ \pi(R|2) & = 0.2; \pi(W|1) = 0.8; \\ \pi(R|2) & = 0.1; \pi(W|2) = 0.9; \\ \pi(R|4) & = 0.9; \pi(W|4) = 0{,}1. \end{cases}$$

III.7 Dans l'exemple du dilemme de l'étudiant (Exemple III.2.16),

(i) Poser sous forme de système, l'équation de Bellman satisfaite par la fonction de valeur optimale v^*.

(ii) Montrer qu'une politique optimale est "$R/W/W/R$".

(iii) Retrouver ce résulta implémentant l'algorithme d'itération de politique 5, puis l'algorithme 6.

III.8 Dans l'exemple du Green Robot (Exemple III.2.2), poser $r_w = 1$, $r_s = 5$, $r_b = 3$, $\gamma = 0{,}9$, $\alpha = 0{,}3$ et $\beta = 0{,}5$.

(i) Calculer la fonction de valeur pour "w/w", "w/s", "w/re", "s/w", "s/s" et "s/re".

(ii) Calculer la fonction de valeur optimale ;

(iii) Montrer qu'une politique optimale est s/re.

(iv) Mêmes question pour $r_w = 4$;

(v) Mêmes questions pour $\alpha = 0{,}9$.

(vi) Retrouver ce résultat en implémentant l'algorithme d'itération de politique 5, puis l'algorithme 6.

III.9 Dans l'exemple AB,

(i) Estimer la fonction de valeur par MC ;

(ii) Estimer la fonction de valeur par TD(0).

Bibliographie

[1] W. Alt, On the approximation of infinite optimization problems with an application to optimal control problems, Appl. Math. Optim. 12 (1984), 15–27.

[2] A.L. Dontchev, Error estimates for a discrete approximation to constrained control problems, SIAM J. Numer. Anal. 18 (1981), 500–514

[3] M. Jonckheere, Introduction to Reinforcement Learning, Note de cours - Centrale Supélec

[4] B. Scherrer, Introduction to Reinforcement Learning, Notes de cours - Univ. Toulouse, 2015

[5] RICHARD S. SUTTON, ANDREW G. BARTO, *Reinforcement Learning : An Introduction*, Second Edition. MIT Press, Cambridge, MA, 2018