Apprentissage par renforcement

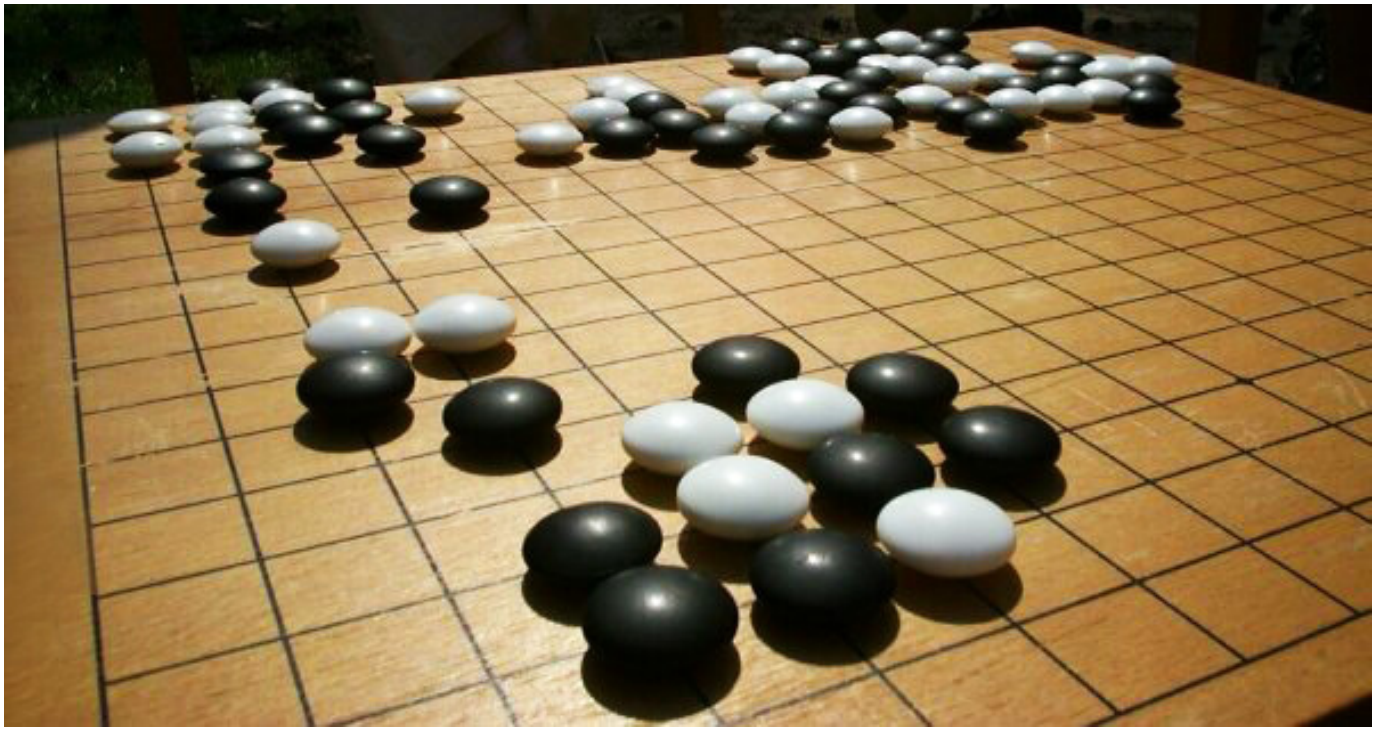
Introduction : Il n'y a pas que la classification...

Le jeu de Go

- Facteur de branchement très important (∼200)
- Difficile de définir une fonction d'évaluation d'une position

⇒ Approches « *arbre minmax* » ne marchent pas

⇒ Peut-on apprendre une bonne stratégie ?



Introduction : Il n'y a pas que la classification...

Un labyrinthe

Comment apprendre à le traverser le plus rapidement possible ?

Ou encore, apprendre une bonne stratégie pour :

- passer l'aspirateur dans une pièce donnée ?
- piloter les ascenseurs d'une tour ?

Introduction : Il n'y a pas que la classification...

Apprentissage par renforcement

= Apprentissage en environnement (partiellement) inconnu

Introduction : Il n'y a pas que la classification...

Apprentissage par renforcement

= Apprentissage en environnement (partiellement) inconnu
= Apprentissage par interaction avec l'environnement (exploration)

Introduction : Il n'y a pas que la classification...

Apprentissage par renforcement

= Apprentissage en environnement (partiellement) inconnu
= Apprentissage par interaction avec l'environnement (exploration)
= Apprentissage par essais et erreurs

Introduction : Il n'y a pas que la classification...

Apprentissage par renforcement

= Apprentissage en environnement (partiellement) inconnu
= Apprentissage par interaction avec l'environnement (exploration)
= Apprentissage par essais et erreurs

Introduction : Il n'y a pas que la classification...

Apprentissage par renforcement

= Apprentissage en environnement (partiellement) inconnu
= Apprentissage par interaction avec l'environnement (exploration)
= Apprentissage par essais et erreurs

≠ Classification, car :

Introduction : Il n'y a pas que la classification...

Apprentissage par renforcement

= Apprentissage en environnement (partiellement) inconnu
= Apprentissage par interaction avec l'environnement (exploration)
= Apprentissage par essais et erreurs

≠ Classification, car :

- l'apprenant peut ***explorer*** les actions possibles («*choisir des exemples*»)

Introduction : Il n'y a pas que la classification...

Apprentissage par renforcement

= Apprentissage en environnement (partiellement) inconnu
= Apprentissage par interaction avec l'environnement (exploration)
= Apprentissage par essais et erreurs

≠ Classification, car :

- l'apprenant peut ***explorer*** les actions possibles (*«choisir des exemples»*)
- ***séquence*** d'actions
  ⇒ la valeur peut n'apparaître qu'à la fin d'une séquence

Introduction : Il n'y a pas que la classification...

Apprentissage par renforcement

= Apprentissage en environnement (partiellement) inconnu
= Apprentissage par interaction avec l'environnement (exploration)
= Apprentissage par essais et erreurs

≠ Classification, car :

- l'apprenant peut ***explorer*** les actions possibles (*«choisir des exemples»*)
- ***séquence*** d'actions
  ⇒ la valeur peut n'apparaître qu'à la fin d'une séquence

Introduction : Il n'y a pas que la classification...

Apprentissage par renforcement

= Apprentissage en environnement (partiellement) inconnu
= Apprentissage par interaction avec l'environnement (exploration)
= Apprentissage par essais et erreurs

≠ Classification, car :

- l'apprenant peut ***explorer*** les actions possibles («*choisir des exemples*»)
- ***séquence*** d'actions
  ⇒ la valeur peut n'apparaître qu'à la fin d'une séquence

≠ Planification

Introduction : Il n'y a pas que la classification...

Apprentissage par renforcement

= Apprentissage en environnement (partiellement) inconnu
= Apprentissage par interaction avec l'environnement (exploration)
= Apprentissage par essais et erreurs

≠ Classification, car :
- l'apprenant peut ***explorer*** les actions possibles («*choisir des exemples*»)
- ***séquence*** d'actions
  ⇒ la valeur peut n'apparaître qu'à la fin d'une séquence

≠ Planification
- car on n'a ici qu'une connaissance ***im***parfaite de l'environnement

Introduction : Il n'y a pas que la classification...

Apprentissage par renforcement

= Apprentissage en environnement (partiellement) inconnu
= Apprentissage par interaction avec l'environnement (exploration)
= Apprentissage par essais et erreurs

≠ Classification, car :
- l'apprenant peut ***explorer*** les actions possibles («*choisir des exemples*»)
- ***séquence*** d'actions
  ⇒ la valeur peut n'apparaître qu'à la fin d'une séquence

≠ Planification
- car on n'a ici qu'une connaissance ***im***parfaite de l'environnement
- car ici l'environnement peut ***changer***

Introduction : Il n'y a pas que la classification...

Apprentissage par renforcement

= Apprentissage en environnement (partiellement) inconnu
= Apprentissage par interaction avec l'environnement (exploration)
= Apprentissage par essais et erreurs

≠ Classification, car :

- l'apprenant peut ***explorer*** les actions possibles (*«choisir des exemples»*)
- ***séquence*** d'actions
  ⇒ la valeur peut n'apparaître qu'à la fin d'une séquence

≠ Planification

- car on n'a ici qu'une connaissance ***im***parfaite de l'environnement
- car ici l'environnement peut ***changer***
- car ici on n'a souvent pas un état initial et un but fixés

Plan du cours et références

- Bandits à plusieurs bras
- Processus de décision markoviens (*MDP*)
- Programmation dynamique
- Apprentissage par renforcement

THE référence : Sutton, R. S. & Barto, A. G. Reinforcement Learning : An Introduction. 2ème édition en préparation, disponible en pdf sur internet. 1ère édition : MIT Press.

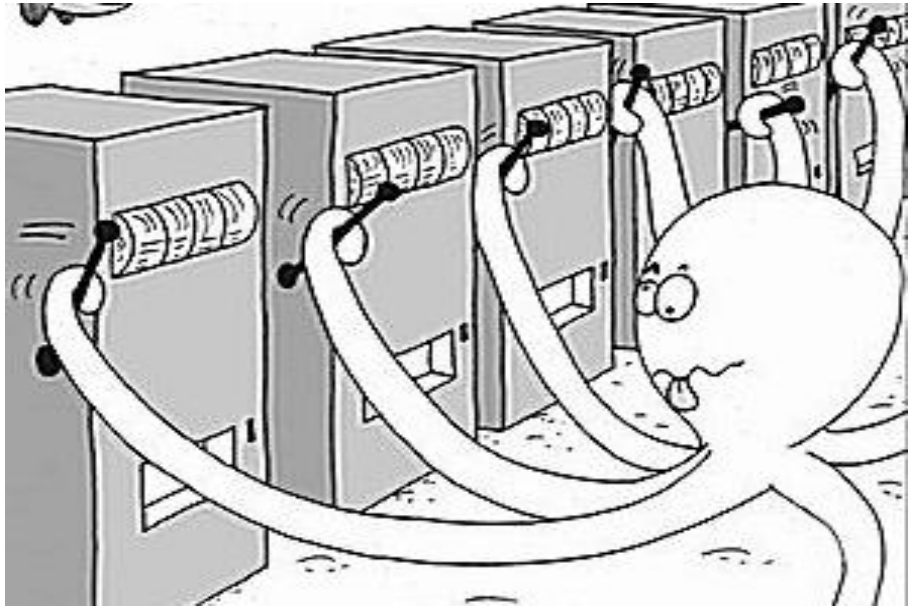
On the WEB : nombreux cours, notamment celui de David Silver : `www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html`

Do you have a favorite coffee place in town ? When you think of having a coffee, you might just go to this place as you're almost sure that you will get the best coffee. But this means [you may be missing an even better coffee elsewhere]....

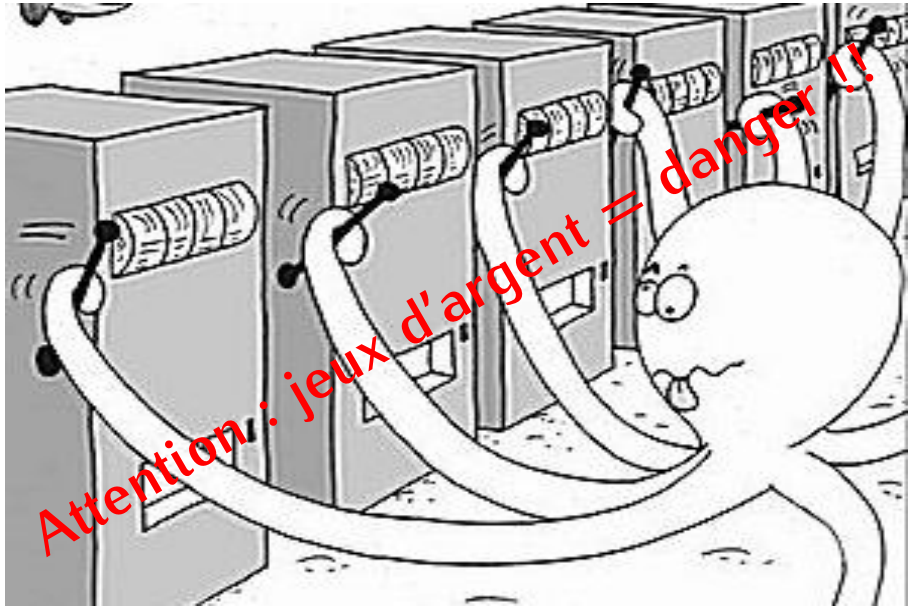
And if you try out all the coffee places one by one, the probability of tasting the worse coffee of your life would be pretty high ! But [...] there's a chance you'll find an even better coffee brewer.

Ankit Choudhary – analyticsvidhya.com, blog, septembre 2018

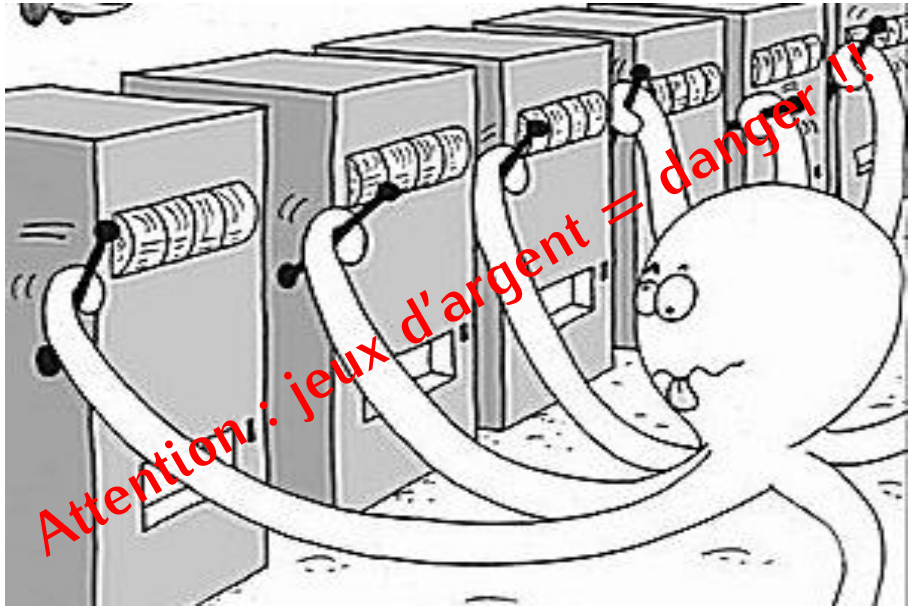
Le bandit à plusieurs bras



Le bandit à plusieurs bras



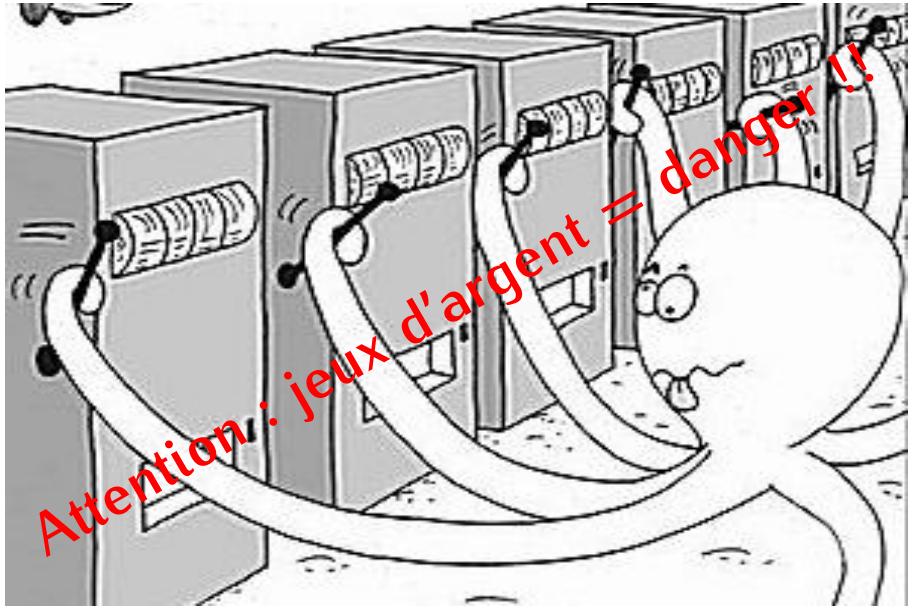
Le bandit à plusieurs bras



- N machines à sous, espérances de gains différentes

$\Rightarrow$ Quelle machine jouer pour maximiser ses gains / minimiser ses pertes

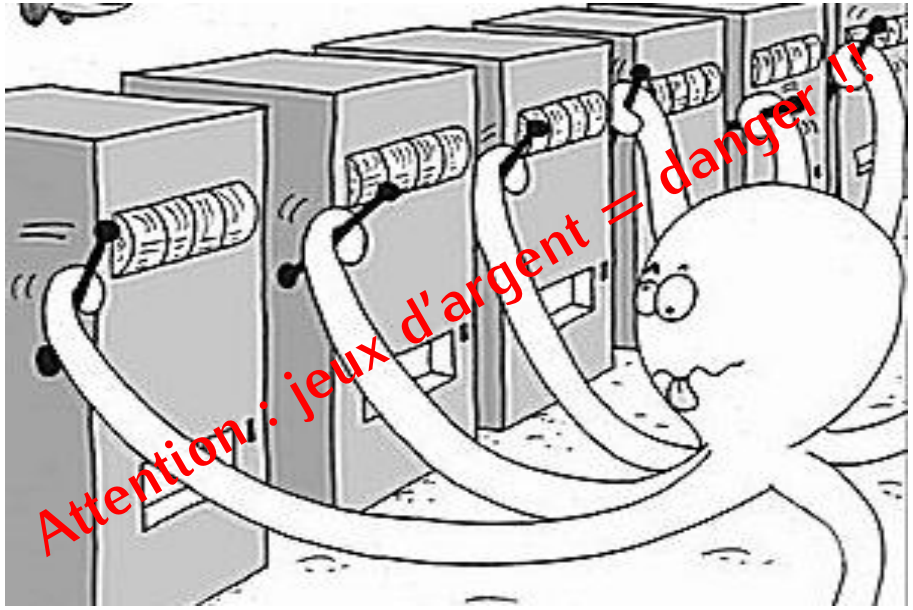
Le bandit à plusieurs bras



Un habitué : connaît les machines !

$\Rightarrow$ Peut jouer la machine qui rapporte le plus en moyenne.

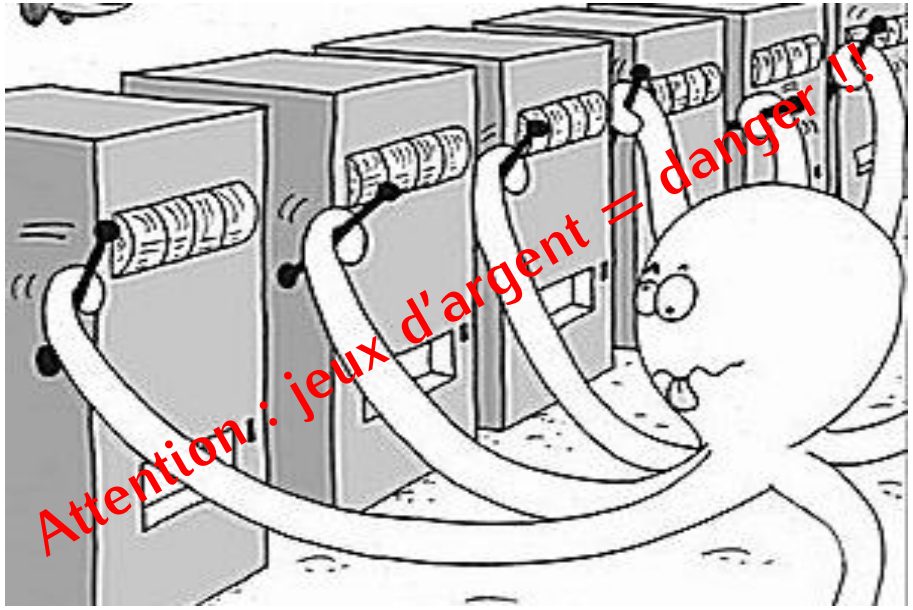
Le bandit à plusieurs bras



Un joueur qui ne connaît pas les machines :

- essayer les machines,

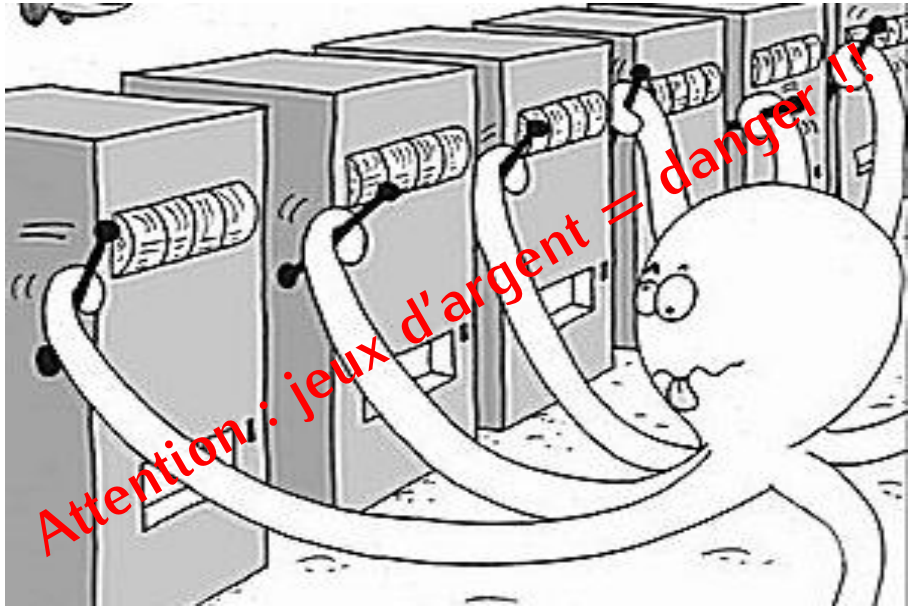
Le bandit à plusieurs bras



Un joueur qui ne connaît pas les machines :

- essayer les machines,
- mais ça peut coûter cher : à chaque partie, on joue 1€

Le bandit à plusieurs bras



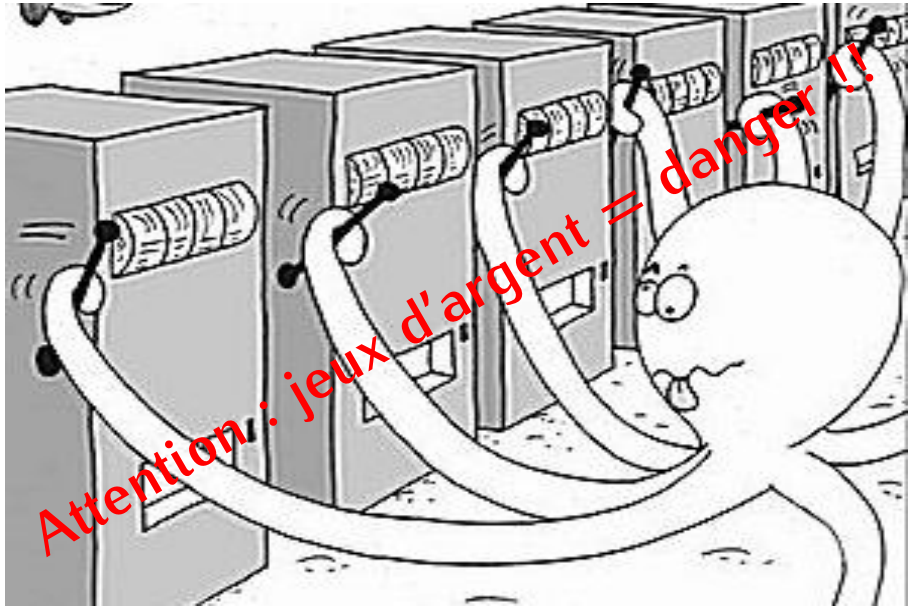
Un joueur qui ne connaît pas les machines :

- essayer les machines,
- mais ça peut coûter cher : à chaque partie, on joue 1€

$\Rightarrow$ quelle stratégie de jeux pour

  - trouver le + vite possible la «*meilleure*» machine
  - en perdant le moins d'argent possible

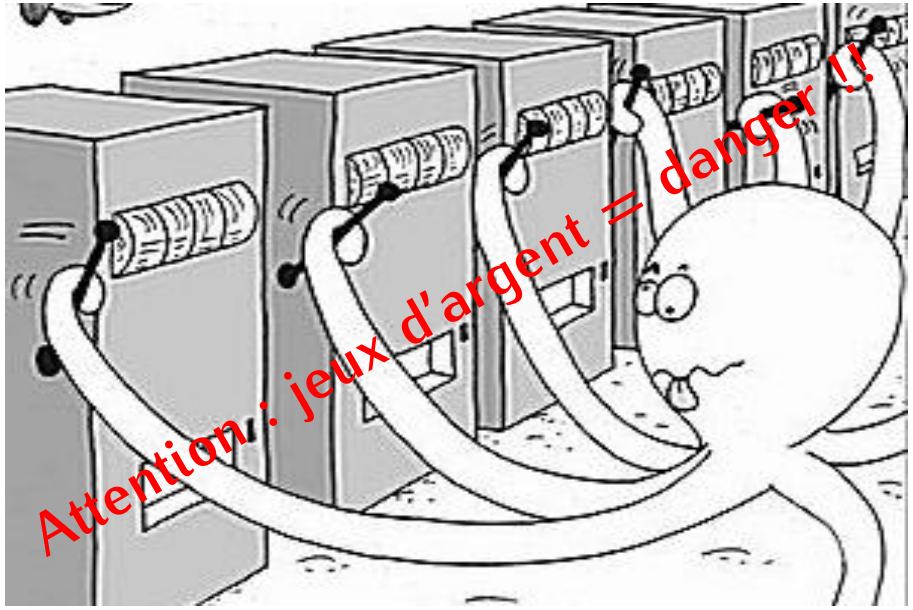
Le bandit à plusieurs bras



Nombreuses applications :

- essais cliniques
- publicités sur pages web
- ...

Le bandit à plusieurs bras



Formalisation du problème

- N machines
- E_k = espérance de gain pour machine k
  $E_k = \sum_{r \in \{-1\} \cup \mathbb{N}} p_k(r) r$
- Les E_k ne changent pas dans le temps
- Les E_k sont inconnues au départ

Le bandit à plusieurs bras

Exemple :

3 machines, avec les gains suivants :

	$R : p(R)$	$R : p(R)$	E
machine 1	200€ : 0,01	-1€ : 0,99	1,01
machine 2	2€ : 0,1	−0,5€ : 0,9	−0,25
machine 3	3€ : 0,1	−0,5€ : 0,9	−0,15

Formalisation du problème

- N machines
- E_k = espérance de gain pour machine k
  $E_k = \sum_{r \in \{-1\} \cup \mathbb{N}} p_k(r) r$
- Les E_k ne changent pas dans le temps
- Les E_k sont inconnues au départ

Le bandit à plusieurs bras : Stratégie «*bornée*»

Exemple :

	$R : p(R)$	$R : p(R)$	E
machine 1	200€ : 0,01	-1€ : 0,99	1,01
machine 2	2€ : 0,1	−0,5€ : 0,9	−0,25
machine 3	3€ : 0,1	−0,5€ : 0,9	−0,15

- On joue m fois chaque machine.
- On choisit ***définitivement*** celle qui a rapporté le meilleur gain.

Le bandit à plusieurs bras : Stratégie «*bornée*»

Exemple :

	$R : p(R)$	$R : p(R)$	E
machine 1	200€ : 0,01	-1€ : 0,99	1,01
machine 2	2€ : 0,1	−0,5€ : 0,9	−0,25
machine 3	3€ : 0,1	−0,5€ : 0,9	−0,15

Si on joue $m = 20$ fois chaque machine :

sur la 1 : 0 fois 200€ $\Rightarrow$ gain moyen $= -20$€

sur la 2 : 3 fois 2€ $\Rightarrow$ gain moyen $= (3 \times 2 - 17 \times 0,5)/20 = -0{,}125$€

sur la 3 : 2 fois 3€ $\Rightarrow$ gain moyen $= (2 \times 3 - 18 \times 0,5)/20 = -0{,}15$€

$\Rightarrow$ On choisit la machine 2

- On joue m fois chaque machine.
- On choisit ***définitivement*** celle qui a rapporté le meilleur gain.

Problème : peut-être que la meilleure machine ***en moyenne*** a donné un faible gain pour les m premiers tours

Le bandit à plusieurs bras : Stratégie «*gourmande*»

- Pas de choix définitif
- À $t+1$-ième partie :
  Choisir la machine la plus prometteuse d'après les t premières parties

Le bandit à plusieurs bras : Stratégie «*gourmande*»

- Pas de choix définitif
- À $t+1$-ième partie :
  Choisir la machine la plus prometteuse d'après les t premières parties

Exemple :

	$R : p(R)$	$R : p(R)$	E
machine 1	200€ : 0,01	-1€ : 0,99	1,01
machine 2	2€ : 0,1	−0,5€ : 0,9	−0,25
machine 3	3€ : 0,1	−0,5€ : 0,9	−0,15

- On commencer par jouer 20 fois chaque machine, on imagine qu'on a :

sur la 1 : 0 fois 200€ $\Rightarrow$ gain moyen $= -1$€
sur la 2 : 3 fois 2€ $\Rightarrow$ gain moyen $= (3 \times 2 - 17 \times 0,5)/20 = -0{,}125$€
sur la 3 : 2 fois 3€ $\Rightarrow$ gain moyen $= (2 \times 3 - 18 \times 0,5)/20 = -0{,}15$€

$\Rightarrow$ On choisit la machine 2

Le bandit à plusieurs bras : Stratégie «*gourmande*»

- Pas de choix définitif
- À $t+1$-ième partie :
  Choisir la machine la plus prometteuse d'après les t premières parties

Exemple :

	$R : p(R)$	$R : p(R)$	E
machine 1	200€ : 0,01	-1€ : 0,99	1,01
machine 2	2€ : 0,1	−0,5€ : 0,9	−0,25
machine 3	3€ : 0,1	−0,5€ : 0,9	−0,15

- On commencer par jouer 20 fois chaque machine, on imagine qu'on a :

sur la 1 : 0 fois 200€ $\Rightarrow$ gain moyen $= -1$€
sur la 2 : 3 fois 2€ $\Rightarrow$ gain moyen $= (3 \times 2 - 17 \times 0,5)/20 = -0{,}125$€
sur la 3 : 2 fois 3€ $\Rightarrow$ gain moyen $= (2 \times 3 - 18 \times 0,5)/20 = -0{,}15$€

$\Rightarrow$ On choisit la machine 2

- En général, après qq dizaines de parties, le gain moyen de 2 va descendre en dessous de −0,15€ (converge vers −0,25)

Le bandit à plusieurs bras : Stratégie «*gourmande*»

Exemple :

	$R : p(R)$	$R : p(R)$	E
machine 1	200€ : 0,01	-1€ : 0,99	1,01
machine 2	2€ : 0,1	−0,5€ : 0,9	−0,25
machine 3	3€ : 0,1	−0,5€ : 0,9	−0,15

- On commencer par jouer 20 fois chaque machine, on imagine qu'on a :

sur la 1 : 0 fois 200€ $\Rightarrow$ gain moyen $= -1$€
sur la 2 : 3 fois 2€ $\Rightarrow$ gain moyen $= (3 \times 2 - 17 \times 0,5)/20 = -0{,}125$€
sur la 3 : 2 fois 3€ $\Rightarrow$ gain moyen $= (2 \times 3 - 18 \times 0,5)/20 = -0{,}15$€

$\Rightarrow$ On choisit la machine 2

- En général, après qq dizaines de parties, le gain moyen de 2 va descendre en dessous de −0,15€ (converge vers −0,25)

$\Rightarrow$ on finira probablement sur la machine 3
mieux que la 2 – mais toujours pas idéal

Le bandit à plusieurs bras : Stratégie «*gourmande*»

- Pas de choix définitif
- À $t+1$-ième partie :
  Choisir la machine la plus prometteuse d'après les t premières parties

Le bandit à plusieurs bras : Estim. des espérances de gains

Nécessité de ***mémoriser*** de l'info sur chaque machine :

$$Q_t(\text{mach}_k) = \frac{\text{somme gains sur machine } k \text{ jusqu'à } t}{\text{nb fois où machine } k \text{ jouée jusqu'à } t}$$

$$\begin{aligned} Q_t(\text{mach}_k) &= \text{valeur estimée de la machine } k \text{ après } t \text{ parties} \\ &= \text{valeur estimée de } E_k \text{ après } t \text{ parties} \end{aligned}$$

Le bandit à plusieurs bras : Estim. des espérances de gains

Nécessité de ***mémoriser*** de l'info sur chaque machine :

$$Q_t(\text{mach}_k) = \frac{\text{somme gains sur machine } k \text{ jusqu'à } t}{\text{nb fois où machine } k \text{ jouée jusqu'à } t}$$

À la $(t+1)$-ième partie :

- on joue la machine k, on observe gain / récompense R_{t+1}
- pour $k' \neq k$: $Q_{t+1}(\text{mach}_{k'}) = Q_t(\text{mach}_{k'})$

Le bandit à plusieurs bras : Estim. des espérances de gains

Nécessité de ***mémoriser*** de l'info sur chaque machine :

$$Q_t(\text{mach}_k) = \frac{\text{somme gains sur machine } k \text{ jusqu'à } t}{\text{nb fois où machine } k \text{ jouée jusqu'à } t}$$

À la $(t+1)$-ième partie :

- on joue la machine k, on observe gain / récompense R_{t+1}
- pour $k' \neq k$: $Q_{t+1}(\text{mach}_{k'}) = Q_t(\text{mach}_{k'})$
- calcul de $Q_{t+1}(\text{mach}_k)$: soit
  - $N_t(\text{mach}_k)$ = nombre de fois où on a joué machine k
  - $\alpha = 1/N_{t+1}(\text{mach}_k)$

Le bandit à plusieurs bras : Estim. des espérances de gains

Nécessité de ***mémoriser*** de l'info sur chaque machine :

$$Q_t(\text{mach}_k) = \frac{\text{somme gains sur machine } k \text{ jusqu'à } t}{\text{nb fois où machine } k \text{ jouée jusqu'à } t}$$

À la $(t+1)$-ième partie :

- on joue la machine k, on observe gain / récompense R_{t+1}
- pour $k' \neq k$: $Q_{t+1}(\text{mach}_{k'}) = Q_t(\text{mach}_{k'})$
- calcul de $Q_{t+1}(\text{mach}_k)$: soit
  - $N_t(\text{mach}_k)$ = nombre de fois où on a joué machine k
  - $\alpha = 1/N_{t+1}(\text{mach}_k)$

  On montre que (cf. exercices) :

$$Q_{t+1}(\text{mach}_k) = (1-\alpha)Q_t(\text{mach}_k) + \alpha R_{t+1}$$

Le bandit à plusieurs bras : Stratégie «*gourmande*»

Apprentissage par renforcement : stratégie gourmande

1. pour tout $k \in 1 \ldots N$: *(Initialisation)*
   a. jouer m partie sur machine k
   b. $N(\text{mach}_k) \leftarrow m$;
   c. $Q(\text{mach}_k) \leftarrow$ le gain moyen sur ces m parties ;

Le bandit à plusieurs bras : Stratégie «*gourmande*»

Apprentissage par renforcement : stratégie gourmande

1. pour tout $k \in 1 \ldots N$: *(Initialisation)*
   a. jouer m partie sur machine k
   b. $N(\text{mach}_k) \leftarrow m$;
   c. $Q(\text{mach}_k) \leftarrow$ le gain moyen sur ces m parties ;
2. tant que *pas fini* : *(Mises à jour)*
   a. choisir k qui maximise $Q(\text{mach}_k)$
   b. jouer machine k, observer récompense R ;
   c. $N(\text{mach}_k) \leftarrow 1 + N(\text{mach}_k)$;
      $Q(\text{mach}_k) \leftarrow (1-\alpha)Q(\text{mach}_k) + \alpha R$;

- $\alpha = 0$: pas de mise à jour (stratégie «*bornée*»)
- $\alpha = 1$: pas de mémoire ;
- $\alpha = \frac{1}{N(\text{mach}_k)}$: les observations comptent de moins en moins

Le bandit à plusieurs bras : Stratégie «*gourmande*»

Apprentissage par renforcement : stratégie gourmande

1. pour tout $k \in 1 \ldots N$: *(Initialisation)*
   a. jouer m partie sur machine k
   b. $N(\text{mach}_k) \leftarrow m$;
   c. $Q(\text{mach}_k) \leftarrow$ le gain moyen sur ces m parties ;
2. tant que *pas fini* : *(Mises à jour)*
   a. choisir k qui maximise $Q(\text{mach}_k)$
   b. jouer machine k, observer récompense R ;
   c. $N(\text{mach}_k) \leftarrow 1 + N(\text{mach}_k)$;
      $Q(\text{mach}_k) \leftarrow (1-\alpha)Q(\text{mach}_k) + \alpha R$;

Problème : on risque d'«*oublier*» une machine qui

- a eu de faibles gains durant l'initialisation, mais
- est pourtant la meilleure.

Autrement dit : aucune garantie de convergence

Le bandit à plusieurs bras : Stratégie ϵ-gourmande

Apprentissage par renforcement : stratégie ϵ-gourmande

1. pour tout $k \in 1 \dots N$: *(Initialisation)*
   a. jouer m partie sur machine k
   b. $N(\text{mach}_k) \leftarrow m$;
   c. $Q(\text{mach}_k) \leftarrow$ le gain moyen sur ces m parties ;
2. tant que *pas fini* : *(Mises à jour)*
   a. $k \leftarrow \begin{cases} \text{argmax}_{k \in 1 \dots N} Q(\text{mach}_k) \text{ avec proba. } 1-\epsilon \\ \text{random}(1 \dots N) \text{ avec proba. } \epsilon \end{cases}$
   b. jouer machine k, observer récompense R ;
   c. $N(\text{mach}_k) \leftarrow 1 + N(\text{mach}_k)$; $\alpha \leftarrow 1/N(\text{mach}_k)$;
   $Q(\text{mach}_k) \leftarrow (1-\alpha)Q(\text{mach}_k) + \alpha R$;

À la $t+1$-ième partie, on choisit

- avec une probabilité $1-\epsilon$: $\text{argmax}_{k \in 1 \dots N} Q_t(\text{mach}_k) =$ exploitation ;
- avec une probabilité ϵ : une machine tirée au hasard = exploration.

Le bandit à plusieurs bras : Stratégie ϵ-gourmande

Apprentissage par renforcement : stratégie ϵ-gourmande

1. pour tout $k \in 1 \dots N$: *(Initialisation)*
   a. jouer m partie sur machine k
   b. $N(\text{mach}_k) \leftarrow m$;
   c. $Q(\text{mach}_k) \leftarrow$ le gain moyen sur ces m parties ;
2. tant que *pas fini* : *(Mises à jour)*
   a. $k \leftarrow \begin{cases} \text{argmax}_{k \in 1 \dots N} Q(\text{mach}_k) \text{ avec proba. } 1-\epsilon \\ \text{random}(1 \dots N) \text{ avec proba. } \epsilon \end{cases}$
   b. jouer machine k, observer récompense R ;
   c. $N(\text{mach}_k) \leftarrow 1 + N(\text{mach}_k)$; $\alpha \leftarrow 1/N(\text{mach}_k)$;
   $Q(\text{mach}_k) \leftarrow (1-\alpha)Q(\text{mach}_k) + \alpha R$;

$\Rightarrow$ Convergence, avec une probabilité de 1, vers la machine ayant la meilleure espérance de gains.

Le bandit à plusieurs bras : Stratégie ϵ-gourmande

Apprentissage par renforcement : stratégie ϵ-gourmande

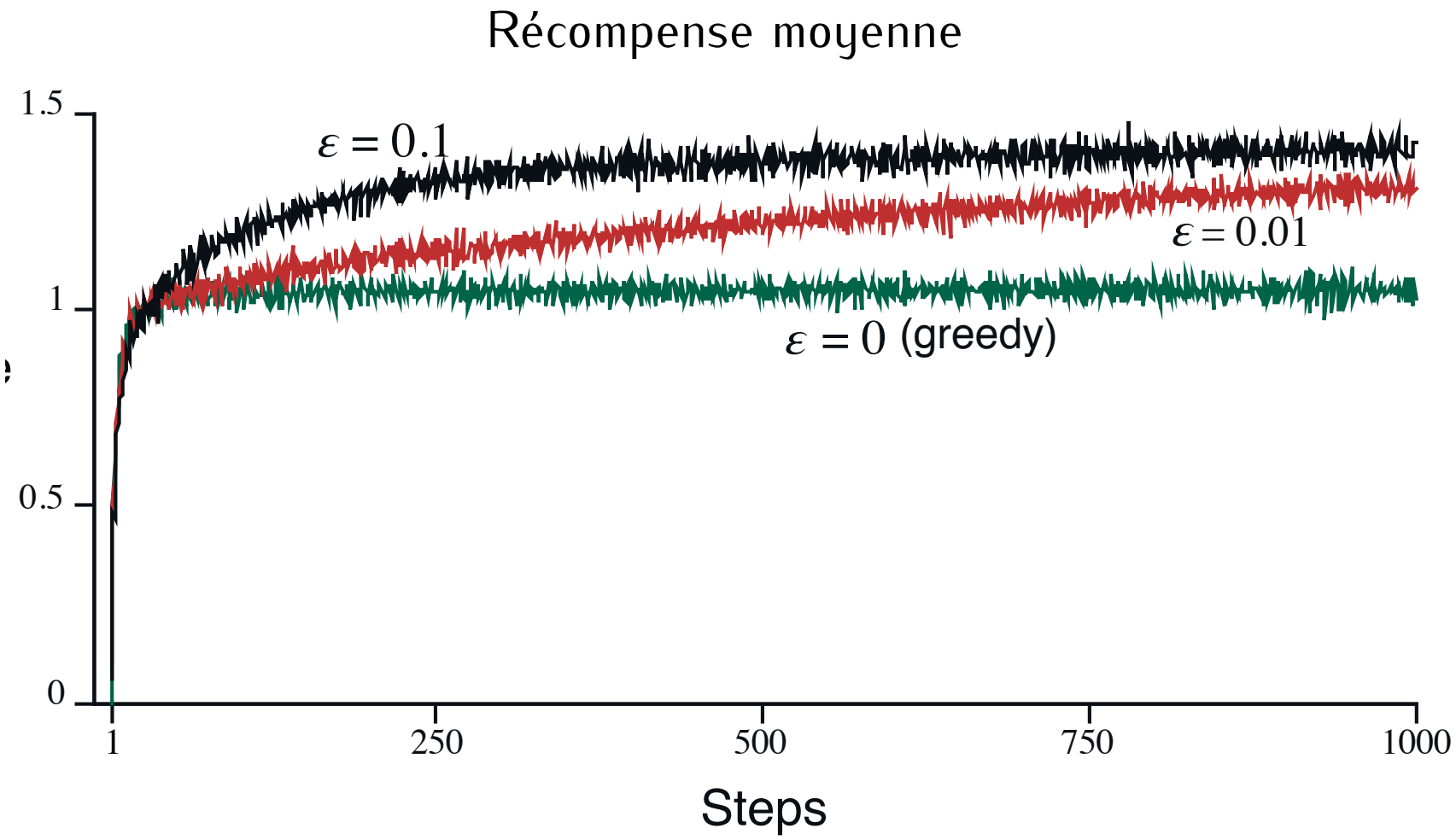
1. pour tout $k \in 1 \ldots N$: *(Initialisation)*
   a. jouer m partie sur machine k
   b. $N(\text{mach}_k) \leftarrow m$;
   c. $Q(\text{mach}_k) \leftarrow$ le gain moyen sur ces m parties ;
2. tant que *pas fini* : *(Mises à jour)*
   a. $k \leftarrow \begin{cases} \text{argmax}_{k \in 1 \ldots N} Q(\text{mach}_k) \text{ avec proba. } 1-\epsilon \\ \text{random}(1 \ldots N) \text{ avec proba. } \epsilon \end{cases}$
   b. jouer machine k, observer récompense R ;
   c. $N(\text{mach}_k) \leftarrow 1 + N(\text{mach}_k)$; $\alpha \leftarrow 1/N(\text{mach}_k)$;
   $Q(\text{mach}_k) \leftarrow (1-\alpha)Q(\text{mach}_k) + \alpha R$;

$\Rightarrow$ Convergence, avec une probabilité de 1, vers la machine ayant la meilleure espérance de gains.

ϵ = compromis entre

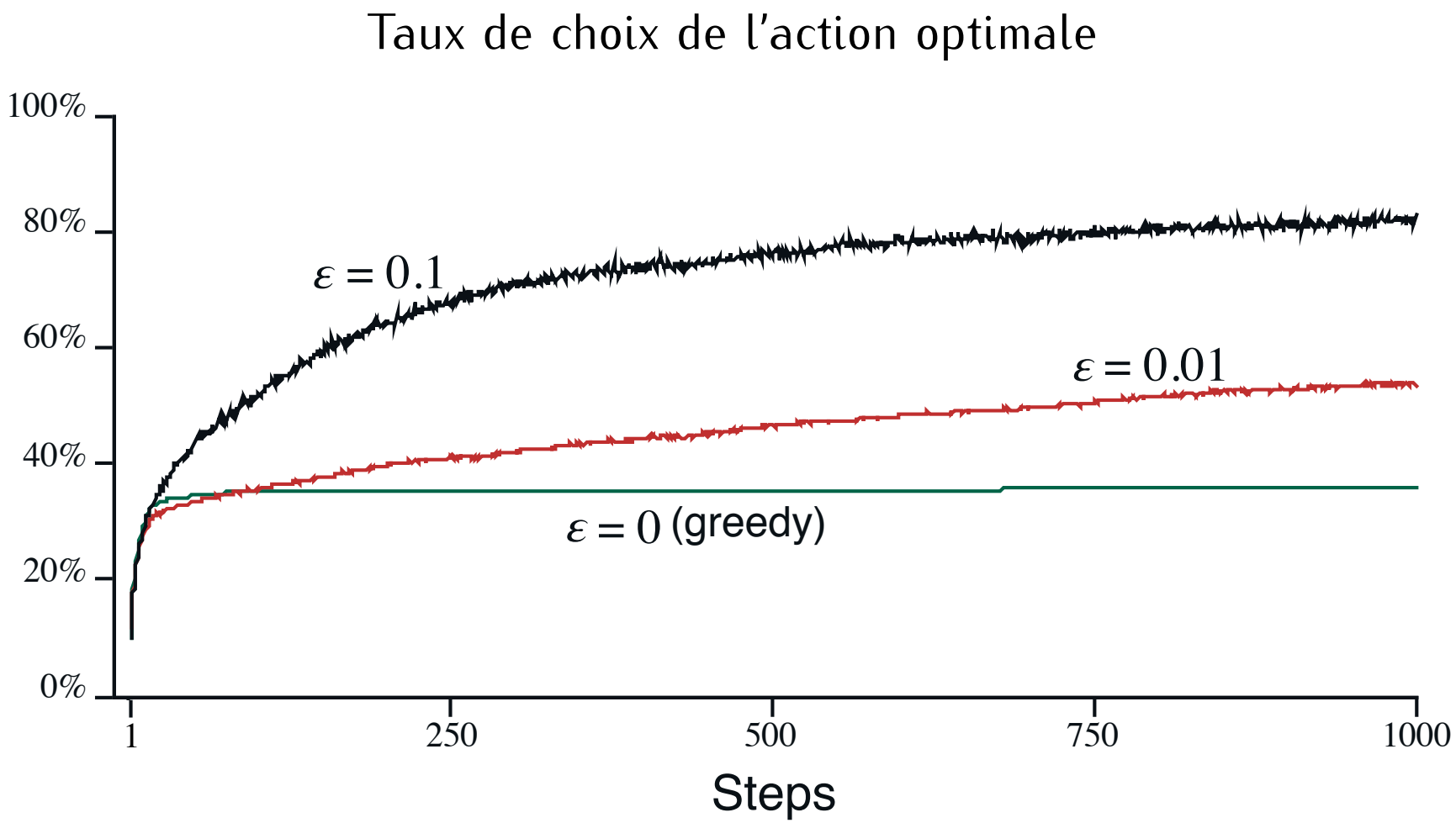
- exploitation de la connaissance déjà acquise
- exploration du système.

Le bandit à plusieurs bras : Expés. – Sutton & Barto



Expérience de Sutton et Barto, moyennes sur 2000 problèmes avec $N = 10$

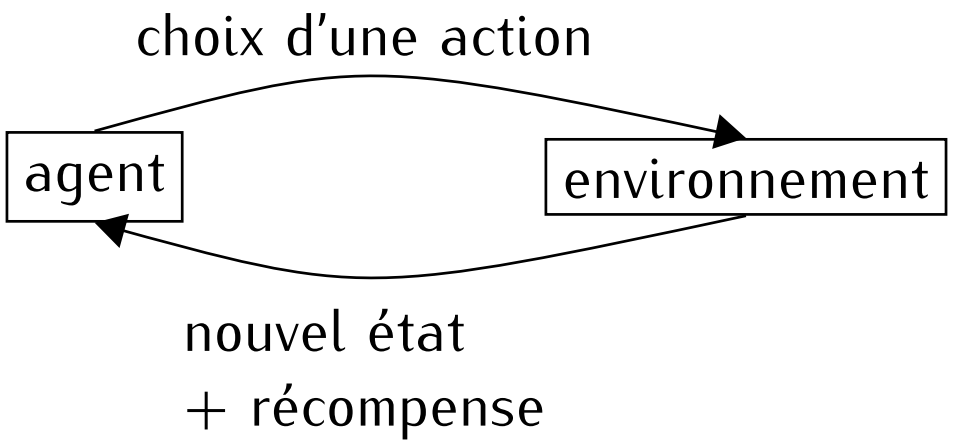
Le bandit à plusieurs bras : Expés. – Sutton & Barto



Remarque

Avec un taux de 0,01 on finit par avoir de meilleurs résultats

Prise de décisions séquentielle



Un agent évolue dans un environnement donné.

Il peut effectuer certaines actions dépendant de l'état courant :

- l'état de l'environnement
- et son propre état

$\Rightarrow$ nouvel état + récompenses / coûts immédiats

L'agent doit apprendre quelle action choisir dans chaque état afin de suivre une séquence d'action qui lui soit la plus favorable possible – en moyenne.

Prise de décisions séquentielle

Jeux :

- chaque coup de l'agent amène une nouvelle situation
- objectif : maximiser les gains, en général ***en fin de partie***
- on ne peut pas apprendre «*la*» bonne séquence :
  le résultat de chaque coup / la nouvelle situation dépend
  - des autres joueurs
  - du hasard pour certains jeux (dés, etc...)

$\Rightarrow$ apprendre le bon coup dans chaque situation

Prise de décisions séquentielle

Apprentissage par renforcement :

permet d'adapter un ***agent*** à un ***environnement***
en tenant compte de ***récompenses***/punitions
(les ***renforcements***)

MDP : États, actions, récompenses

MDP = Processus de décision Markovien = un modèle :

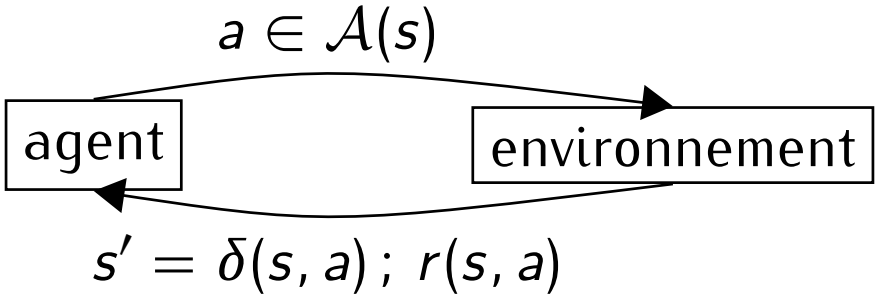
$\mathcal{S}$ un ensemble d'états

$\mathcal{A}$ un ensemble d'actions

$\mathcal{A}(s)$ ensemble des actions réalisables dans l'état s

$r(s,a) \in \mathbb{R}$ récompense immédiate

$\delta(s,a) \in \mathcal{S}$ état résultant



MDP : États, actions, récompenses

MDP = Processus de décision Markovien = un modèle :

$\mathcal{S}$ un ensemble d'états

$\mathcal{A}$ un ensemble d'actions

$\mathcal{A}(s)$ ensemble des actions réalisables dans l'état s

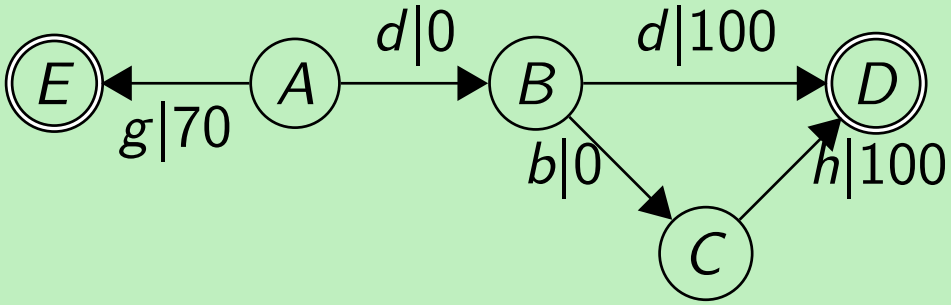
$r(s, a) \in \mathbb{R}$ récompense immédiate

$\delta(s, a) \in \mathcal{S}$ état résultant

Exemple :

5 états, 4 actions (g,d,h,b)

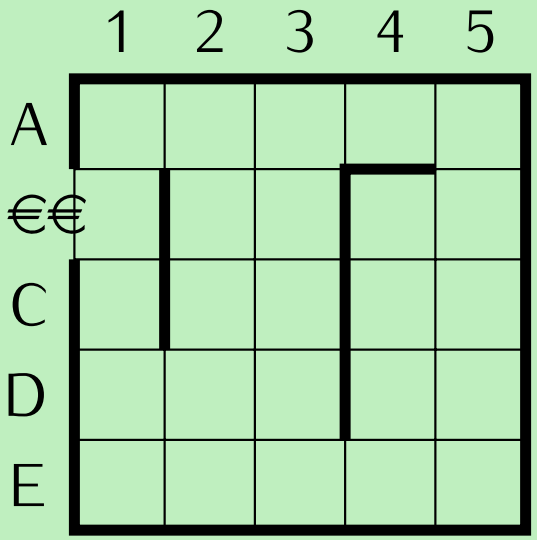
$r(A, g) = 70, \ldots$



MDP : États, actions, récompenses

Exemple : Un agent dans un labyrinthe

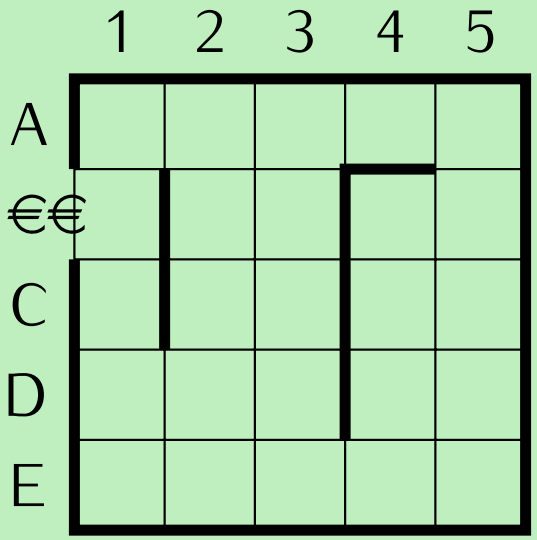
- Déplacements possibles : $\uparrow \downarrow \leftarrow \rightarrow$ (sauf à travers les murs)
- Chaque déplacement coûte 5€
- Gain à la sortie : 100€



MDP : États, actions, récompenses

Exemple : Un agent dans un labyrinthe

- Déplacements possibles : $\uparrow \downarrow \leftarrow \rightarrow$ (sauf à travers les murs)
- Chaque déplacement coûte 5€
- Gain à la sortie : 100€

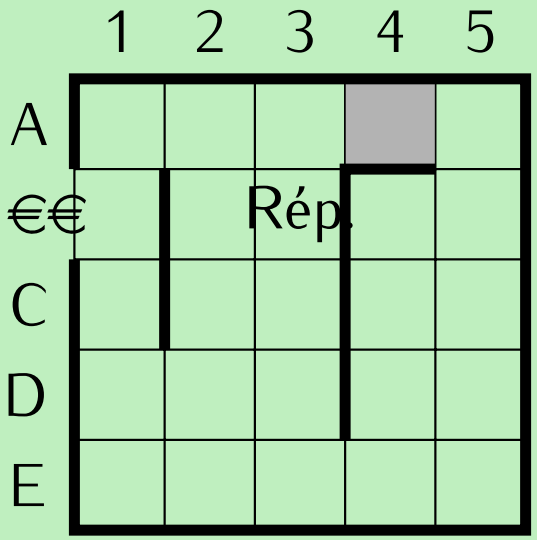


Modèle : 25 états, 4 actions

MDP : États, actions, récompenses

Exemple : Un agent dans un labyrinthe

- Déplacements possibles : $\uparrow \downarrow \leftarrow \rightarrow$
  (sauf à travers les murs)
- Chaque déplacement coûte 5€
- Gain à la sortie : 100€ / 50€ si blessé
- Piège en (A,4)
  Réparation immédiate en (B,3)

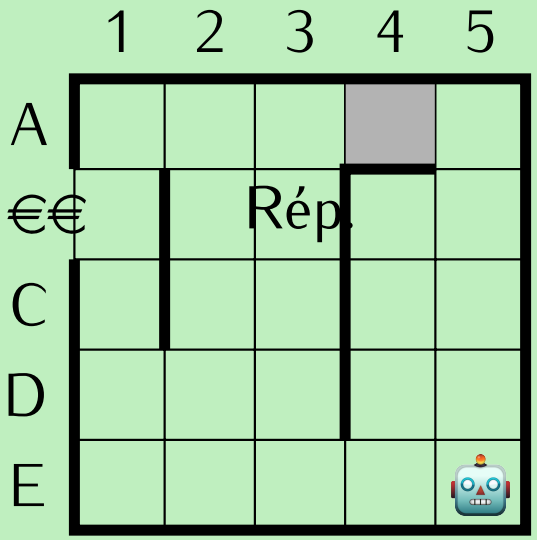


Modèle : 50 états, 4 actions

MDP : États, actions, récompenses

Exemple : Un agent dans un labyrinthe

- Déplacements possibles : $\uparrow \downarrow \leftarrow \rightarrow$
  (sauf à travers les murs)
- Chaque déplacement coûte 5€
- Gain à la sortie : 100€ / 50€ si blessé
- Piège en (A,4)
  Réparation immédiate en (B,3)



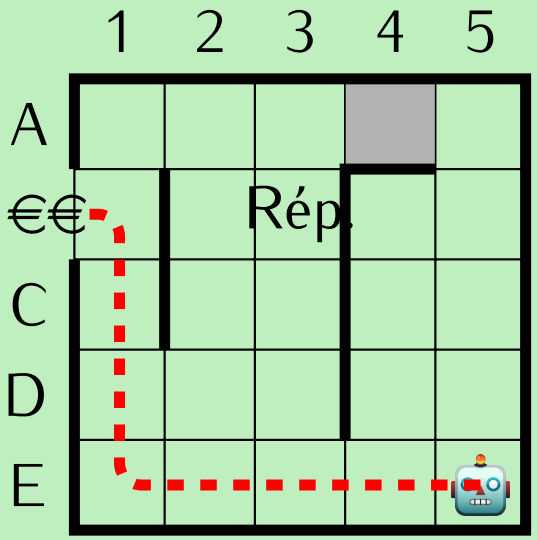
Modèle : 50 états, 4 actions

- Meilleure séquence d'actions à partir de (E,5) ?

MDP : États, actions, récompenses

Exemple : Un agent dans un labyrinthe

- Déplacements possibles : $\uparrow \downarrow \leftarrow \rightarrow$
  (sauf à travers les murs)
- Chaque déplacement coûte 5€
- Gain à la sortie : 100€ / 50€ si blessé
- Piège en (A,4)
  Réparation immédiate en (B,3)



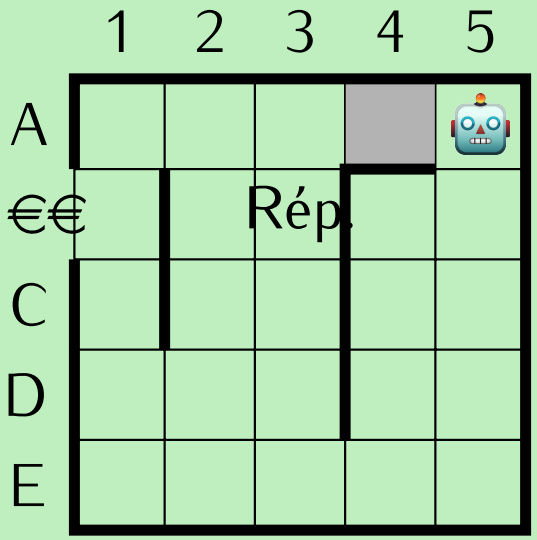
Modèle : 50 états, 4 actions

- Meilleure séquence d'actions à partir de (E,5) ?

$$\leftarrow, \leftarrow, \leftarrow, \leftarrow, \uparrow, \uparrow, \uparrow, \leftarrow \Rightarrow \text{gain} = 100 - 8 \times 5 = 60€$$

MDP : États, actions, récompenses

Exemple : Un agent dans un labyrinthe



- Déplacements possibles : $\uparrow \downarrow \leftarrow \rightarrow$
  (sauf à travers les murs)
- Chaque déplacement coûte 5€
- Gain à la sortie : 100€ / 50€ si blessé
- Piège en (A,4)
  Réparation immédiate en (B,3)

Modèle : 50 états, 4 actions

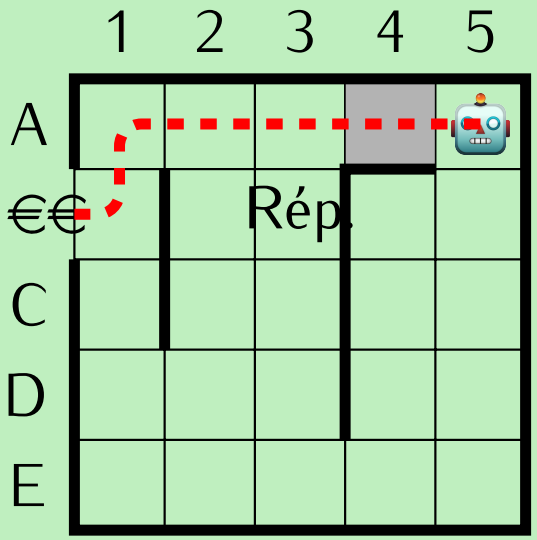
- Meilleure séquence d'actions à partir de (E,5) ?

$$\leftarrow, \leftarrow, \leftarrow, \leftarrow, \uparrow, \uparrow, \uparrow, \leftarrow \Rightarrow \text{gain} = 100 - 8 \times 5 = 60€$$

- Meilleure séquence d'actions à partir de (A,5) ?

MDP : États, actions, récompenses

Exemple : Un agent dans un labyrinthe



- Déplacements possibles : $\uparrow \downarrow \leftarrow \rightarrow$
  (sauf à travers les murs)
- Chaque déplacement coûte 5€
- Gain à la sortie : 100€ / 50€ si blessé
- Piège en (A,4)
  Réparation immédiate en (B,3)

Modèle : 50 états, 4 actions

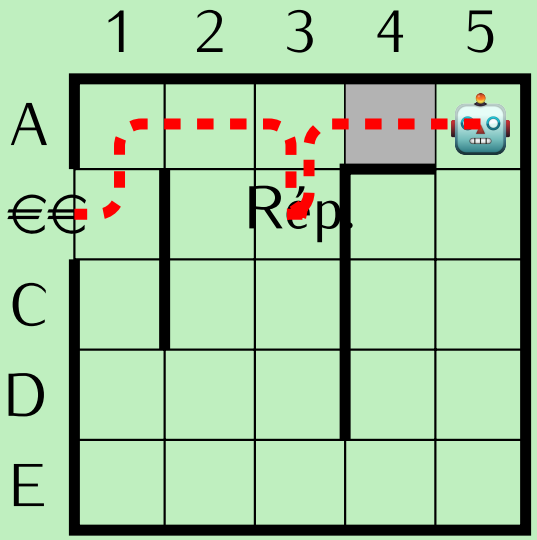
- Meilleure séquence d'actions à partir de (E,5) ?

  $\leftarrow, \leftarrow, \leftarrow, \leftarrow, \uparrow, \uparrow, \uparrow, \leftarrow \Rightarrow \text{gain} = 100 - 8 \times 5 = 60€$
- Meilleure séquence d'actions à partir de (A,5) ?

  $\uparrow, \leftarrow, \leftarrow, \leftarrow, \leftarrow, \downarrow, \leftarrow \Rightarrow \text{gain} = 50 - 5 \times 6 = 20€$

MDP : États, actions, récompenses

Exemple : Un agent dans un labyrinthe



- Déplacements possibles : $\uparrow \downarrow \leftarrow \rightarrow$
  (sauf à travers les murs)
- Chaque déplacement coûte 5€
- Gain à la sortie : 100€ / 50€ si blessé
- Piège en (A,4)
  Réparation immédiate en (B,3)

Modèle : 50 états, 4 actions

- Meilleure séquence d'actions à partir de (E,5) ?

  $\leftarrow, \leftarrow, \leftarrow, \leftarrow, \uparrow, \uparrow, \uparrow, \leftarrow \Rightarrow \text{gain} = 100 - 8 \times 5 = 60€$
- Meilleure séquence d'actions à partir de (A,5) ?

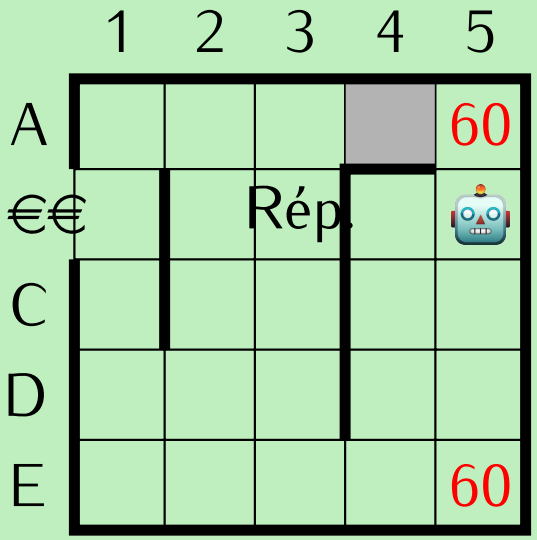
  $\uparrow, \leftarrow, \leftarrow, \leftarrow, \leftarrow, \downarrow, \leftarrow \Rightarrow \text{gain} = 50 - 5 \times 6 = 20€$

  $\uparrow, \leftarrow, \leftarrow, \downarrow, \uparrow, \leftarrow, \leftarrow, \downarrow, \leftarrow \Rightarrow \text{gain} = 100 - 5 \times 8 = 60€$

MDP : États, actions, récompenses

Exemple : Un agent dans un labyrinthe

- Déplacements possibles : $\uparrow \downarrow \leftarrow \rightarrow$
  (sauf à travers les murs)
- Chaque déplacement coûte 5€
- Gain à la sortie : 100€ / 50€ si blessé
- Piège en (A,4)
  Réparation immédiate en (B,3)



Modèle : 50 états, 4 actions

- Meilleure séquence d'actions à partir de (E,5) ?

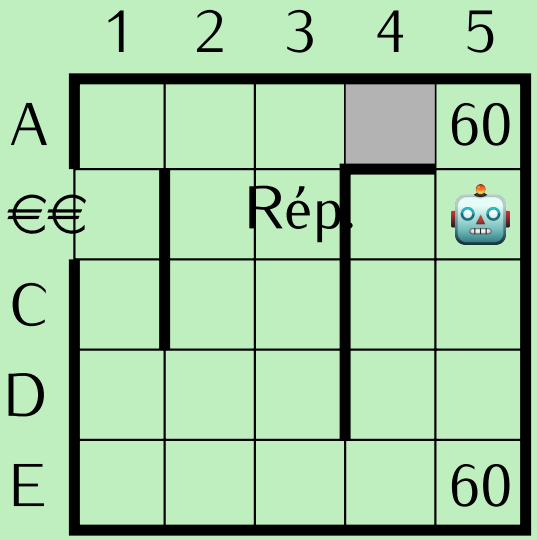
  $\leftarrow, \leftarrow, \leftarrow, \leftarrow, \uparrow, \uparrow, \uparrow, \leftarrow \Rightarrow \text{gain} = 100 - 8 \times 5 = 60€$
- Meilleure séquence d'actions à partir de (A,5) ?

  $\uparrow, \leftarrow, \leftarrow, \leftarrow, \leftarrow, \downarrow, \leftarrow \Rightarrow \text{gain} = 50 - 5 \times 6 = 20€$

  $\uparrow, \leftarrow, \leftarrow, \downarrow, \uparrow, \leftarrow, \leftarrow, \downarrow, \leftarrow \Rightarrow \text{gain} = 100 - 5 \times 8 = 60€$
- Meilleure séquence d'actions à partir de (B,5) ?

MDP : États, actions, récompenses

Exemple : Un agent dans un labyrinthe



- Déplacements possibles : $\uparrow \downarrow \leftarrow \rightarrow$
  (sauf à travers les murs)
- Chaque déplacement coûte 5€
- Gain à la sortie : 100€ / 50€ si blessé
- Piège en (A,4)
  Réparation immédiate en (B,3)

Modèle : 50 états, 4 actions

- Meilleure séquence d'actions à partir de (E,5) ?
  $\leftarrow, \leftarrow, \leftarrow, \leftarrow, \uparrow, \uparrow, \uparrow, \leftarrow \Rightarrow$gain $= 100 - 8 \times 5 = 60$€
- Meilleure séquence d'actions à partir de (A,5) ?
  $\uparrow, \leftarrow, \leftarrow, \leftarrow, \leftarrow, \downarrow, \leftarrow \Rightarrow$ gain $= 50 - 5 \times 6 = 20$€
  $\uparrow, \leftarrow, \leftarrow, \downarrow, \uparrow, \leftarrow, \leftarrow, \downarrow, \leftarrow \Rightarrow$ gain $= 100 - 5 \times 8 = 60$€
- Meilleure séquence d'actions à partir de (B,5) ?
  par (A,4) : $60 - 1 = 55$€ ; par (E,5) : $60 - 3 \times 5 = 45$€

MDP : Qualité d'une action et optimalité

Récompense associée à une séquence états / actions :

$$\mathrm{s} = s_0 \longrightarrow a_0 \rightarrow s_1 \rightarrow a1 \rightarrow s_2 \rightarrow \ldots$$

$$\Rightarrow R(\mathrm{s}) = r(s_0, a_0) + \quad r(s_1, a_1) + \quad r(s_2, a_2) + \ldots = \sum_{i=0}^{+\infty} r(s_i, a_i)$$

MDP : Qualité d'une action et optimalité

Récompense associée à une séquence états / actions :

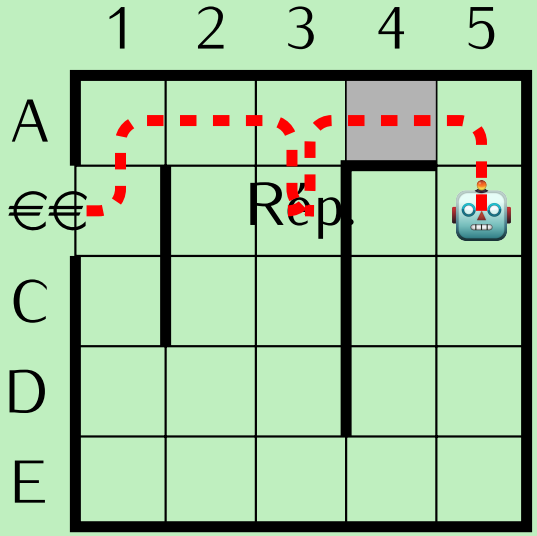
$$s = s_0 \longrightarrow a_0 \rightarrow s_1 \rightarrow a1 \rightarrow s_2 \rightarrow \ldots$$

$$\Rightarrow R(s) = r(s_0, a_0) + \quad r(s_1, a_1) + \quad r(s_2, a_2) + \ldots = \sum_{i=0}^{+\infty} r(s_i, a_i)$$

Exemple : le labyrinthe – suite

$$R((B, 5, \neg\text{blessé}), \uparrow, \leftarrow, \leftarrow, \downarrow, \uparrow, \leftarrow, \leftarrow, \downarrow, \leftarrow)$$
$$= 100 - 9 \times 5 = 55 \text{ €}$$

$$R((B, 5, \neg\text{blessé}), \uparrow, \leftarrow, \leftarrow, \leftarrow, \leftarrow, \downarrow, \leftarrow)$$
$$= 50 - 7 \times 5 = 15 \text{ €}$$



MDP : Qualité d'une action et optimalité

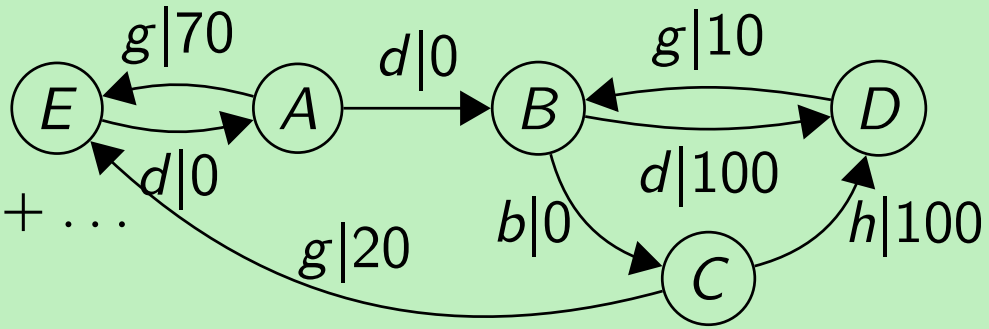
Récompense associée à une séquence états / actions :

$$s = s_0 \longrightarrow a_0 \rightarrow s_1 \rightarrow a1 \rightarrow s_2 \rightarrow \ldots$$

$$\Rightarrow R(s) = r(s_0, a_0) + \quad r(s_1, a_1) + \quad r(s_2, a_2) + \ldots = \sum_{i=0}^{+\infty} r(s_i, a_i)$$

Exemple :

$$R(A, g, d, g, d, g, d, \ldots)$$
$$= 70 + \quad 0 + \quad 70 + \quad 0 + \ldots$$



MDP : Qualité d'une action et optimalité

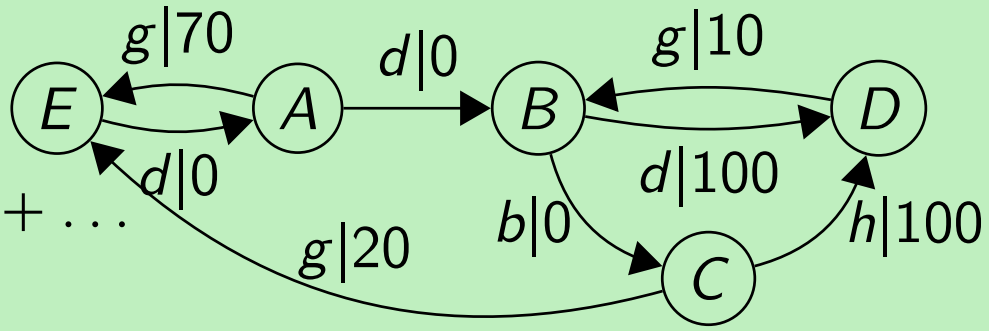
Récompense associée à une séquence états / actions :

$$s = s_0 \longrightarrow a_0 \rightarrow s_1 \rightarrow a1 \rightarrow s_2 \rightarrow \ldots$$

$$\Rightarrow R(s) = r(s_0, a_0) + \quad r(s_1, a_1) + \quad r(s_2, a_2) + \ldots = \sum_{i=0}^{+\infty} r(s_i, a_i)$$

Exemple :

$$R(A, g, d, g, d, g, d, \ldots)$$
$$= 70 + \quad 0 + \quad 70 + \quad 0 + \ldots$$
$$= +\infty \ldots$$



MDP : Qualité d'une action et optimalité

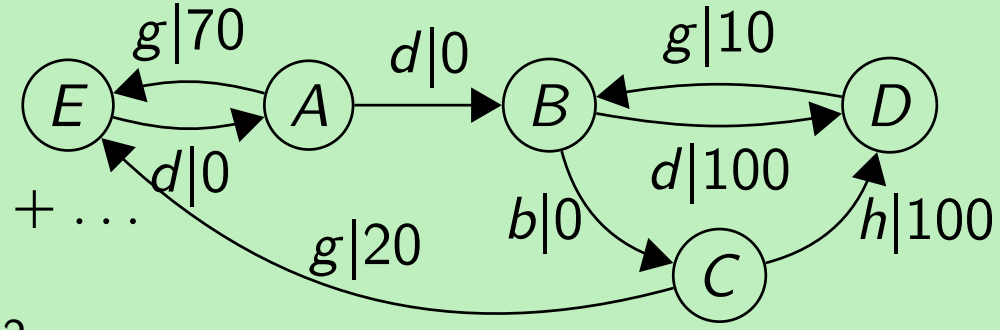
Récompense associée à une séquence états / actions :

$$s = s_0 \longrightarrow a_0 \rightarrow s_1 \rightarrow a1 \rightarrow s_2 \rightarrow \ldots$$

$$\Rightarrow R(s) = r(s_0, a_0) + \gamma r(s_1, a_1) + \gamma^2 r(s_2, a_2) + \ldots = \sum_{i=0}^{+\infty} \gamma^i r(s_i, a_i)$$

Exemple :

$$R(A, g, d, g, d, g, d, \ldots)$$
$$= 70 + \gamma 0 + \gamma^2 70 + \gamma^3 0 + \ldots$$
$$= 70 \sum_{i=0}^{+\infty} \gamma^{2i} = 70/(1-\gamma^2)$$



Pour garantir la convergence de la série : on fixe $\gamma \in]0, 1[$.

MDP : Qualité d'une action et optimalité

Qualité d'une action a : dans un état donné s, tel que $a \in \mathcal{A}(s)$

$$Q^*(s, a) = \max R(\mathrm{s})$$

où on considère les séquences s d'états / actions :

- qui commencent par s, a
- qui vont jusqu'à un état terminal
- ou sont infinies (s'il y a des cycles)

MDP : Qualité d'une action et optimalité

Qualité d'une action a : dans un état donné s, tel que $a \in \mathcal{A}(s)$

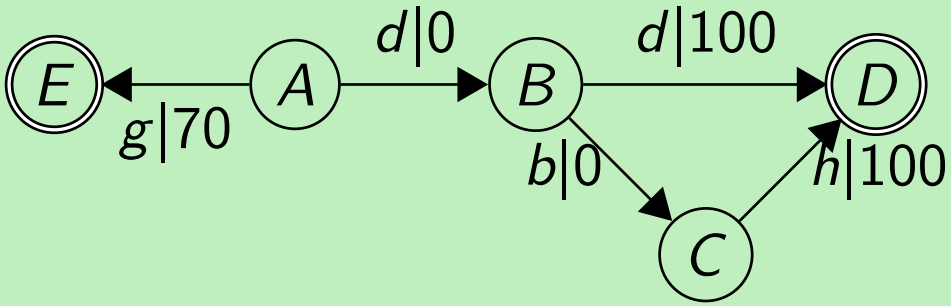
$$Q^*(s, a) = \max R(\mathrm{s})$$

où on considère les séquences s d'états / actions :

- qui commencent par s, a
- qui vont jusqu'à un état terminal
- ou sont infinies (s'il y a des cycles)

Exemple :

Avec γ=0,8 :
$Q^*(C,h)=R(C,h)=100$



MDP : Qualité d'une action et optimalité

Qualité d'une action a : dans un état donné s, tel que $a \in \mathcal{A}(s)$

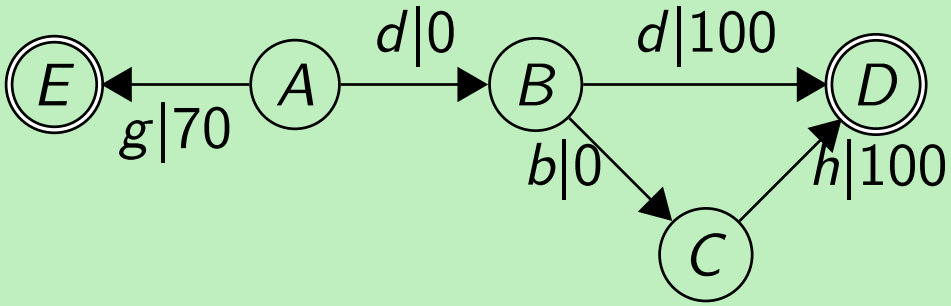
$$Q^*(s, a) = \max R(\mathrm{s})$$

où on considère les séquences s d'états / actions :

- qui commencent par s, a
- qui vont jusqu'à un état terminal
- ou sont infinies (s'il y a des cycles)

Exemple :

Avec γ=0,8 :
$Q^*(C,h)=R(C,h)=100$
$Q^*(B,b)=R(B,b,h)=100\gamma=80$



MDP : Qualité d'une action et optimalité

Qualité d'une action a : dans un état donné s, tel que $a \in \mathcal{A}(s)$

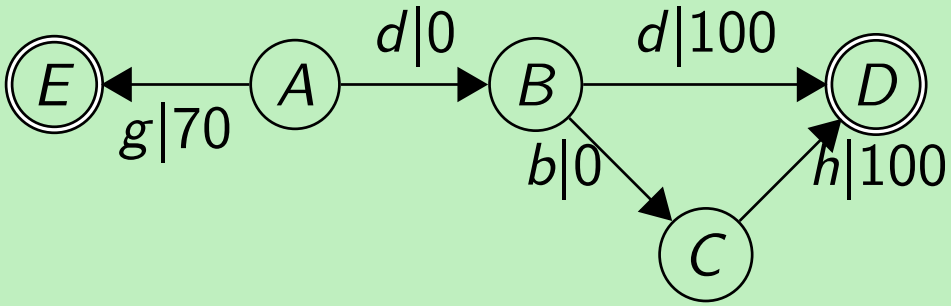
$$Q^*(s, a) = \max R(\mathrm{s})$$

où on considère les séquences s d'états / actions :

- qui commencent par s, a
- qui vont jusqu'à un état terminal
- ou sont infinies (s'il y a des cycles)

Exemple :

Avec γ=0,8 :
$Q^*(C,h)=R(C,h)=100$
$Q^*(B,b)=R(B,b,h)=100\gamma=80$
$Q^*(B,d)=R(B,d)=100$



MDP : Qualité d'une action et optimalité

Qualité d'une action a : dans un état donné s, tel que $a \in \mathcal{A}(s)$

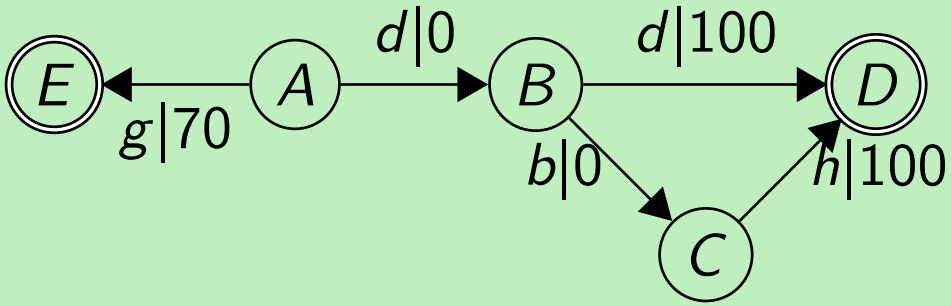
$$Q^*(s, a) = \max R(\mathrm{s})$$

où on considère les séquences s d'états / actions :

- qui commencent par s, a
- qui vont jusqu'à un état terminal
- ou sont infinies (s'il y a des cycles)

Exemple :

Avec γ=0,8 :
$Q^*(C,h)=R(C,h)=100$
$Q^*(B,b)=R(B,b,h)=100\gamma=80$
$Q^*(B,d)=R(B,d)=100$
$Q^*(A,d)=R(A,d,d)=100\gamma=80$



MDP : Qualité d'une action et optimalité

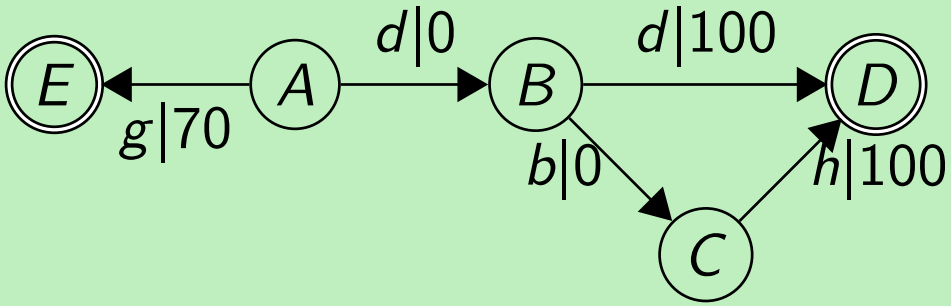
Qualité d'une action a : dans un état donné s, tel que $a \in \mathcal{A}(s)$

$$Q^*(s, a) = \max R(\mathrm{s})$$

où on considère les séquences s d'états / actions :

- qui commencent par s, a
- qui vont jusqu'à un état terminal
- ou sont infinies (s'il y a des cycles)

Exemple :



Avec γ=0,8 :
$Q^*(C,h)=R(C,h)=100$
$Q^*(B,b)=R(B,b,h)=100\gamma=80$
$Q^*(B,d)=R(B,d)=100$
$Q^*(A,d)=R(A,d,d)=100\gamma=80$
$\Rightarrow Q^*(A, d) = 0 + \gamma \max\{Q^*(B, b), Q^*(B, d)\}$

MDP : Qualité d'une action et optimalité

Qualité d'une action a : dans un état donné s, tel que $a \in \mathcal{A}(s)$

$$Q^*(s, a) = \max R(\mathrm{s})$$

où on considère les séquences s d'états / actions :

- qui commencent par s, a
- qui vont jusqu'à un état terminal
- ou sont infinies (s'il y a des cycles)

Quand on connaît tous les $Q^*(s, a)$, pour toute $a \in \mathcal{A}(s)$:

- pour maximiser le gain à partir de s,
  on choisit a qui maximise $Q^*(s, a)$.

MDP : Qualité d'une action et optimalité

Qualité d'une action a : dans un état donné s, tel que $a \in \mathcal{A}(s)$

$$Q^*(s, a) = \max R(\mathrm{s})$$

où on considère les séquences s d'états / actions :

- qui commencent par s, a
- qui vont jusqu'à un état terminal
- ou sont infinies (s'il y a des cycles)

Propriété (Équations de Bellman)

Pour tous $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$, en notant $s_1 = \delta(s, a)$:

$$Q^*(s, a) = r(s, a) + \gamma \max_{a_1 \in \mathcal{A}(s_1)} Q^*(s_1, a_1)$$

Problème : équations non linéaires ! *(max.)*

MDP : Qualité d'une action et optimalité

Qualité d'une action a : dans un état donné s, tel que $a \in \mathcal{A}(s)$

$$Q^*(s, a) = \max R(\mathrm{s})$$

Résoudre le MDP = calculer les $Q^*(s, a)$ pour tous s, a

Une fois qu'on connaît les $Q^*(s, a)$:

on peut, dans un état s, choisir une action a qui a un $Q^*(s, a)$ max.

MDP : Politique

Une ***politique*** π formalise le choix des actions :

$$\pi : \mathcal{S} \to \mathcal{A} \qquad \text{telle que } \pi(s) \in \mathcal{A}(s)$$

Une politique dit quoi faire dans chaque état.

MDP : Politique

Une ***politique*** π formalise le choix des actions :

$$\pi : \mathcal{S} \to \mathcal{A} \qquad \text{telle que } \pi(s) \in \mathcal{A}(s)$$

Une politique dit quoi faire dans chaque état.

Exemple : le labyrinthe – suite

Politique 1 :

- vers le Nord si possible ; sinon
- vers l'Est si possible ; sinon
- vers l'Ouest si possible ; sinon
- vers le Sud

Politique 2 :

	1	2	3	4	5	
A	↓	←	←/↓	←	←	
B	←	↓	↑	↓	↑	⇐
C	↑	↓	↓	↓	↓	⇐
D	↑	←	←	↓	↓	⇐
E	↑	←	←	←	←	
		⇑	⇑	⇑		

Rép.

Professeur Tournesol : «*toujours plus à l'Ouest*»

MDP : Politique

Une ***politique*** π formalise le choix des actions :

$$\pi : \mathcal{S} \to \mathcal{A} \qquad \text{telle que } \pi(s) \in \mathcal{A}(s)$$

Une politique dit quoi faire dans chaque état.

Pour π et état s_0 donnés :

- une séquence unique

$$s_0 \to a_0 = \pi(s_0) \to s_1 = \delta(s_0, a_0) \to a1 = \pi(s1) \to s_2 = \delta(s1, a1) \to \dots$$

MDP : Politique

Une ***politique*** π formalise le choix des actions :

$$\pi : \mathcal{S} \to \mathcal{A} \qquad \text{telle que } \pi(s) \in \mathcal{A}(s)$$

Une politique dit quoi faire dans chaque état.

Pour π et état s_0 donnés :

- une séquence unique

$$s_0 \to a_0 = \pi(s_0) \to s_1 = \delta(s_0, a_0) \to a1 = \pi(s1) \to s_2 = \delta(s1, a1) \to \ldots$$

- une valeur unique :

$$R(\pi, s_0) = r(s_0, a_0) + \quad r(s_1, a_1) + \quad r(s_2, a_2) + \ldots = \sum_{i=0}^{+\infty} \quad r(s_i, a_i)$$

MDP : Politique

Une ***politique*** π formalise le choix des actions :

$$\pi : \mathcal{S} \to \mathcal{A} \qquad \text{telle que } \pi(s) \in \mathcal{A}(s)$$

Une politique dit quoi faire dans chaque état.

Pour π et état s_0 donnés :

- une séquence unique

$$s_0 \to a_0 = \pi(s_0) \to s_1 = \delta(s_0, a_0) \to a1 = \pi(s1) \to s_2 = \delta(s1, a1) \to \dots$$

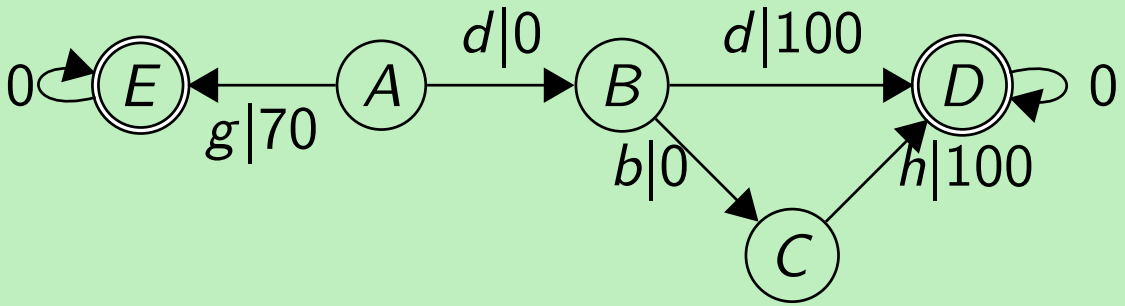
- une valeur unique :

$$R(\pi, s_0) = r(s_0, a_0) + \gamma r(s_1, a_1) + \gamma^2 r(s_2, a_2) + \dots = \sum_{i=0}^{+\infty} \gamma^i r(s_i, a_i)$$

Pour garantir la convergence de la série : on fixe $\gamma \in]0, 1[$.

MDP : Politique

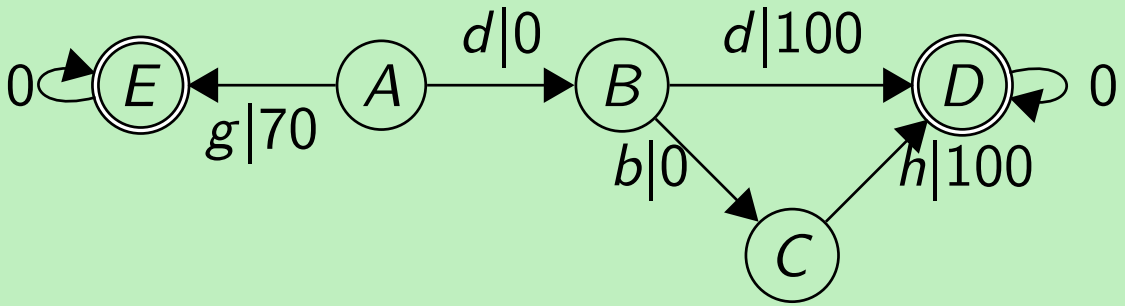
Exemple :



$\pi_1(A) = d$, $\pi_1(B) = b$, $\gamma = 0,8$

MDP : Politique

Exemple :

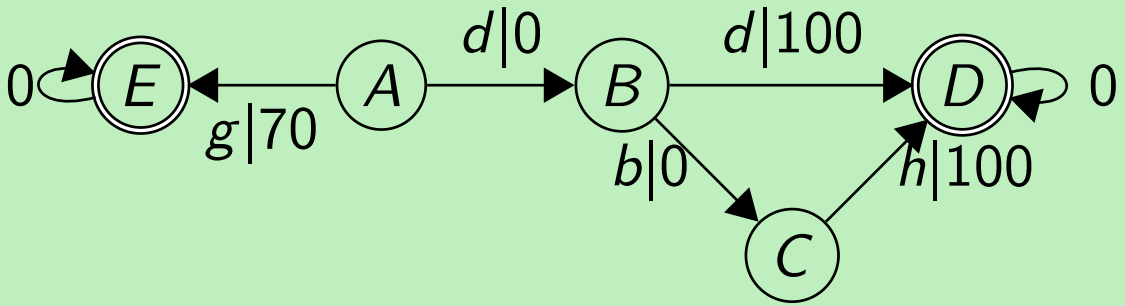


$\pi_1(A) = d$, $\pi_1(B) = b$, $\gamma = 0,8$

$\Rightarrow R(\pi_1, A) = 0 + \gamma \times 0 + \gamma^2 \times 100 = 64$

MDP : Politique

Exemple :



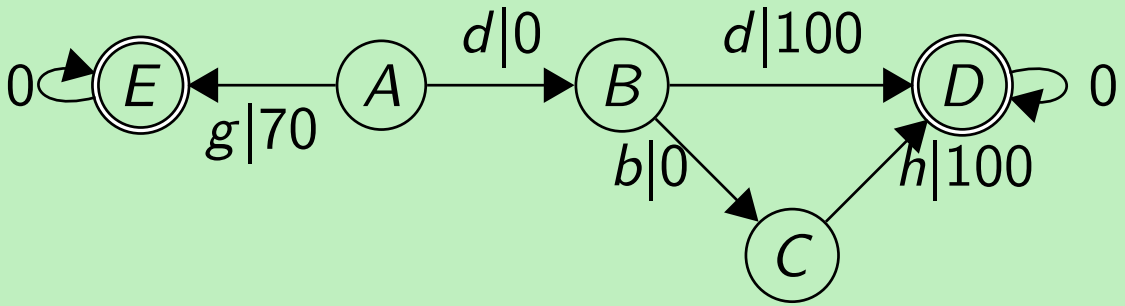
$\pi_1(A) = d$, $\pi_1(B) = b$, $\gamma = 0,8$

$\Rightarrow R(\pi_1, A) = 0 + \gamma \times 0 + \gamma^2 \times 100 = 64$

$\pi_2(A) = g$, $\pi_2(B) = d$, $\gamma = 0,8$

MDP : Politique

Exemple :



$\pi_1(A) = d,\ \pi_1(B) = b,\ \gamma = 0,8$

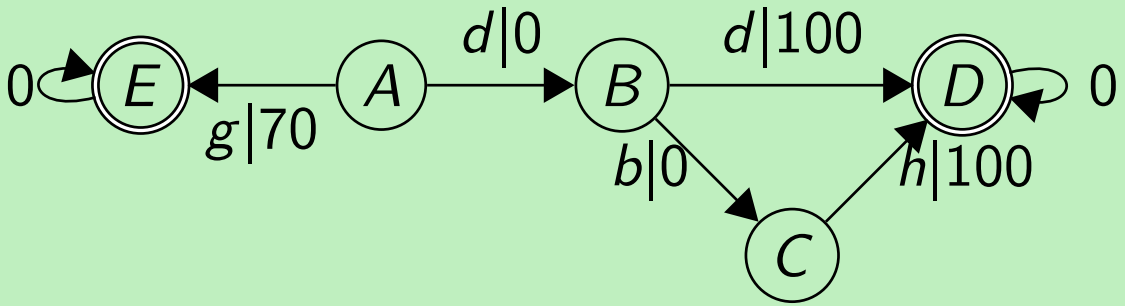
$\Rightarrow\ R(\pi_1, A) = 0 + \gamma \times 0 + \gamma^2 \times 100 = 64$

$\pi_2(A) = g,\ \pi_2(B) = d,\ \gamma = 0,8$

$\Rightarrow\ R(\pi_2, A) = 70$

MDP : Politique

Exemple :



$\pi_1(A) = d,\ \pi_1(B) = b,\ \gamma = 0,8$

$\Rightarrow\ R(\pi_1, A) = 0 + \gamma \times 0 + \gamma^2 \times 100 = 64$

$\pi_2(A) = g,\ \pi_2(B) = d,\ \gamma = 0,8$

$\Rightarrow\ R(\pi_2, A) = 70$

MDP : Politique

On introduit aussi Q :

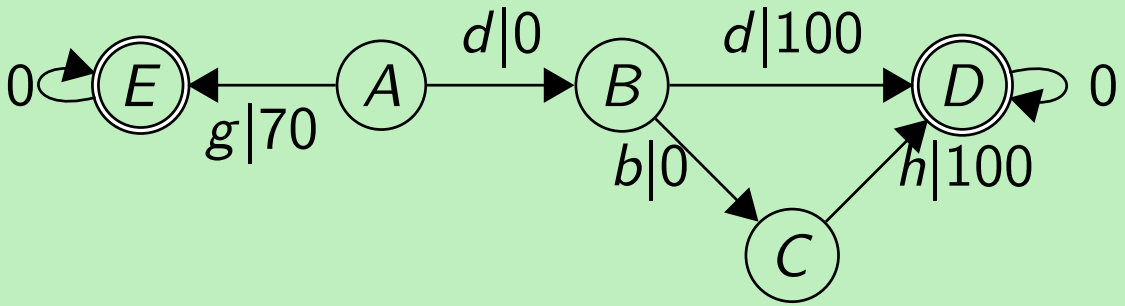
$$\begin{aligned} Q(\pi, s, a) &= r(s, a) + \gamma r(s_1, \pi(s_1)) + \gamma^2 r(s_2, \pi(s_2)) + \dots \\ &= r(s, a) + \gamma R(\pi, \delta(s, a)) \end{aligned}$$

Q permet de détailler ce qui se passe si :

- on fixe la première action a ;
- les actions suivantes sont dictées par π

MDP : Politique

Exemple :



$\pi_1(A) = d,\ \pi_1(B) = b,\ \gamma = 0,8$

$\Rightarrow\ R(\pi_1, A) = 0 + \gamma \times 0 + \gamma^2 \times 100 = 64$

$\pi_2(A) = g,\ \pi_2(B) = d,\ \gamma = 0,8$

$\Rightarrow\ R(\pi_2, A) = 70$

$Q(\pi_1, A, g) = 70$

MDP : Politique

Politique $\neq$ stratégie :

- une politique donne une information très «*locale*» : que jouer dans un état donné
- nous parlerons plus tard de «*stratégie d'apprentissage*»

MDP : Optimalité

- Politique optimale π

$$R(\pi, s) \geq R(\pi', s) \text{ pour tous } s, \pi'$$

MDP : Optimalité

- Politique optimale π

$$R(\pi, s) \geq R(\pi', s) \text{ pour tous } s, \pi'$$

- Si $\mathcal{S}$, $\mathcal{A}$ finis : il existe au moins une politique optimale π^*

MDP : Optimalité

- Politique optimale π

$$R(\pi, s) \geq R(\pi', s) \text{ pour tous } s, \pi'$$

- Si $\mathcal{S}$, $\mathcal{A}$ finis : il existe au moins une politique optimale π^*

Theorem (Système d'équations vérifié par les politiques optimales)

Si π^* est optimale, alors pour tous $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$,
en notant $s_1 = \delta(s, a)$:

$$Q(\pi^*, s, a) = r(s, a) + \gamma \max_{a' \in \mathcal{A}(s_1)} Q(\pi^*, s_1, a')$$

Mais ça n'est pas un système d'équations linéaires ordinaire (max)
$\Rightarrow$ pas aussi facile à résoudre...

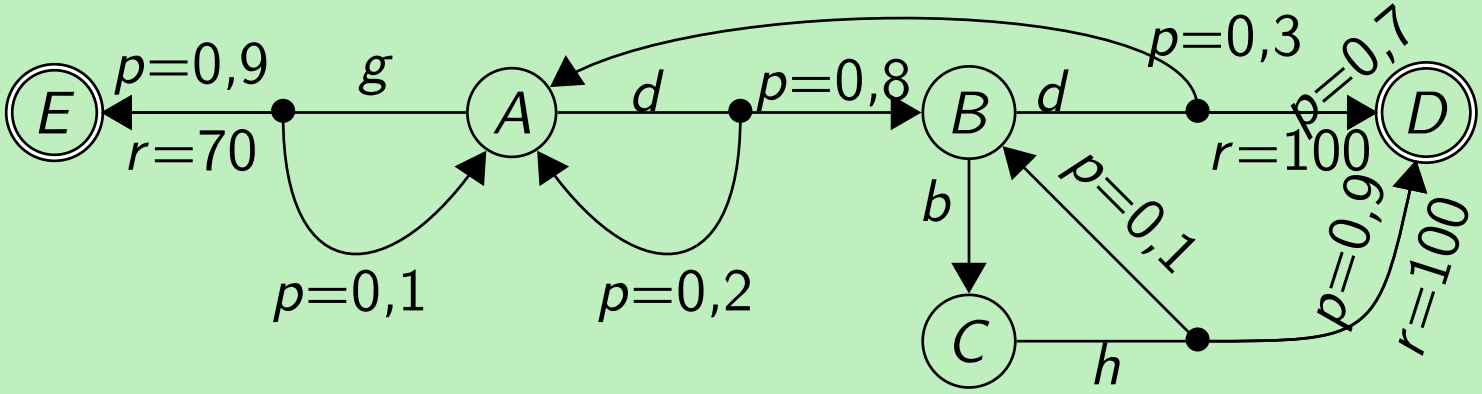
MDP : Cas non déterministe

Dans de nombreuses situations, le résultat d'une action ne peut être prédit avec certitude. Par exemple :

- Jeux à plusieurs joueurs
  $\Rightarrow$ le résultat d'un coup d'un joueur dépend des coups des autres
- Robot en déplacement
  $\Rightarrow$ le résultat d'une accélération dépend de l'adhérence
  ...

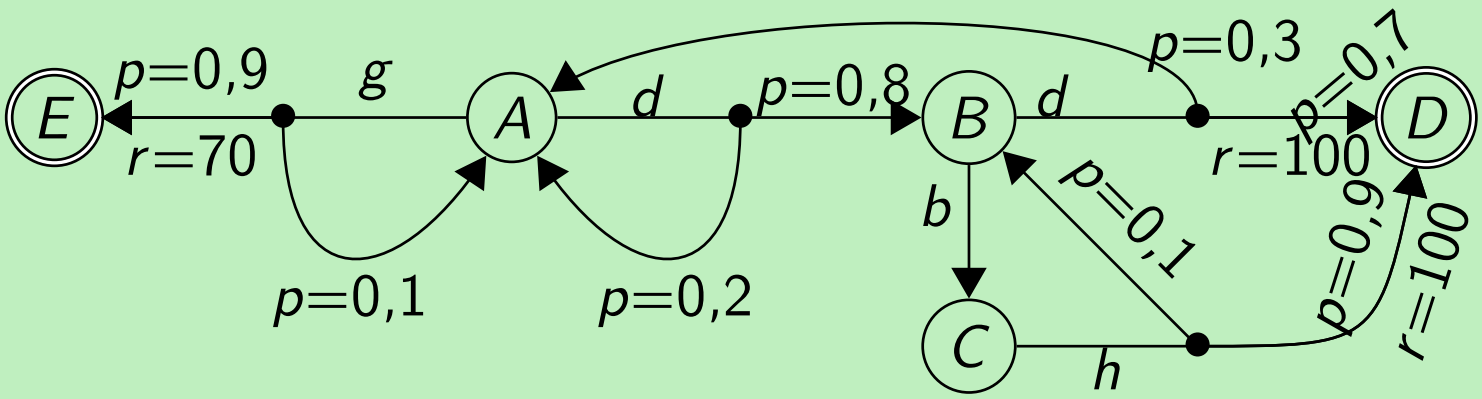
MDP : Cas non déterministe

Exemple :



MDP : Cas non déterministe

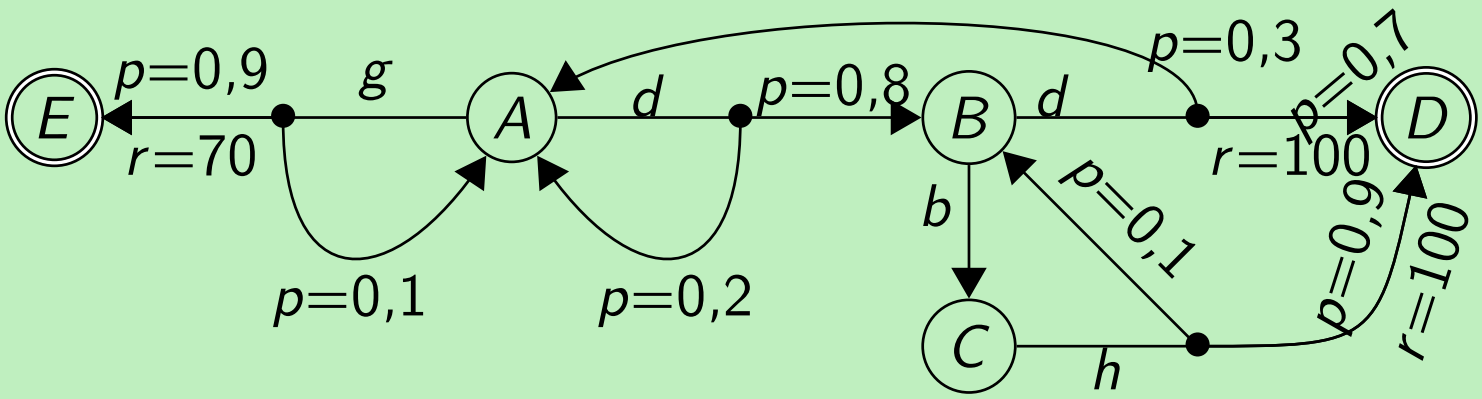
Exemple :



- action d en A :
  - avec une probabilité $0,8$: on arrive en B, récompense nulle
  - avec une probabilité $0,2$: on reste en A, récompense nulle

MDP : Cas non déterministe

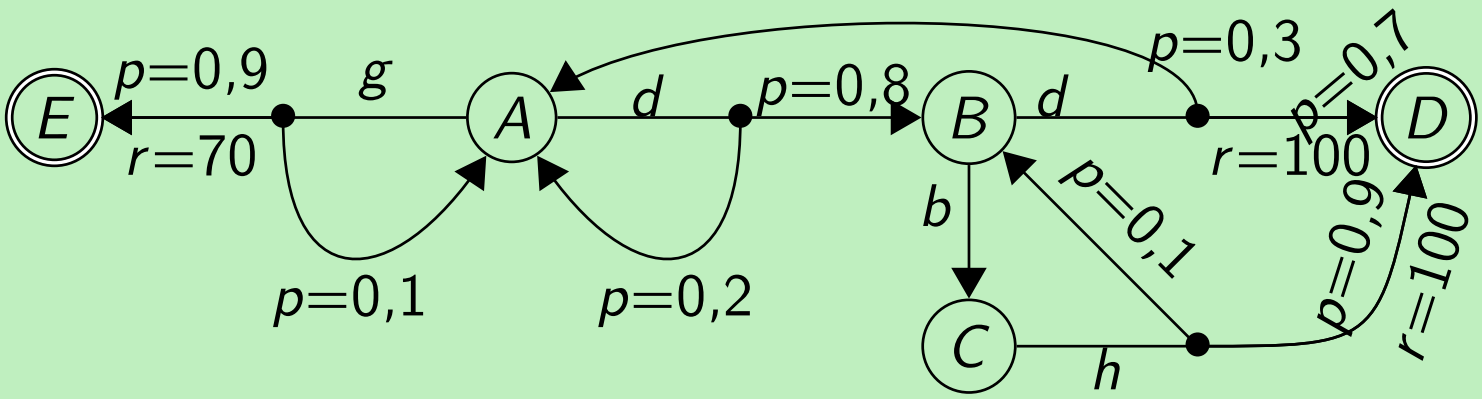
Exemple :



- action d en A :
  - avec une probabilité $0,8$: on arrive en B, récompense nulle
  - avec une probabilité $0,2$: on reste en A, récompense nulle
- action h en C :
  - avec une probabilité $0,9$: on arrive en D, récompense 100
  - avec une probabilité $0,1$: on reste en C, récompense nulle

MDP : Cas non déterministe

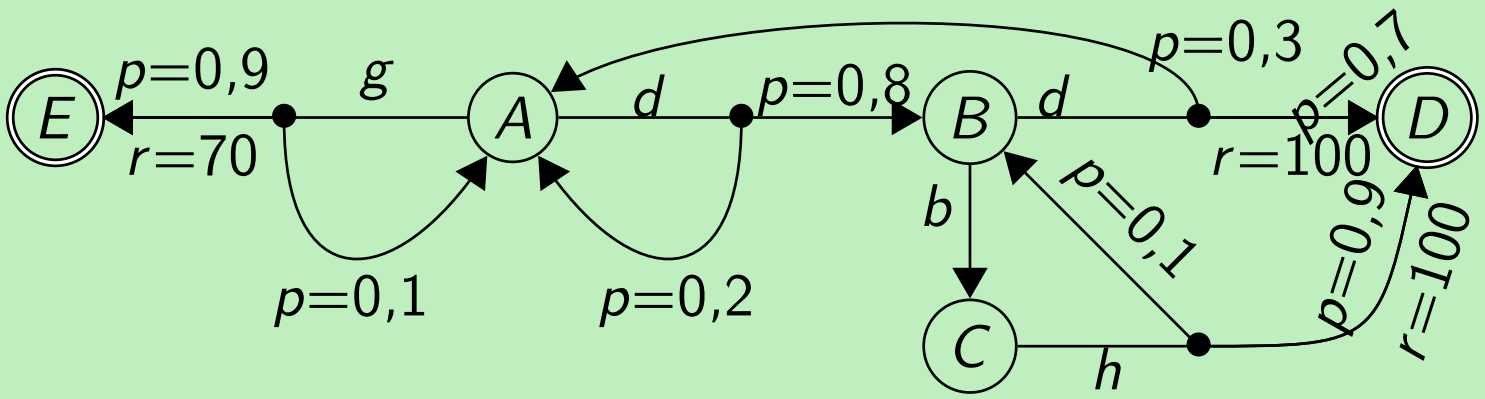
Exemple :



- séquence A, g, E : $R = 70$, $p = 0,9$

MDP : Cas non déterministe

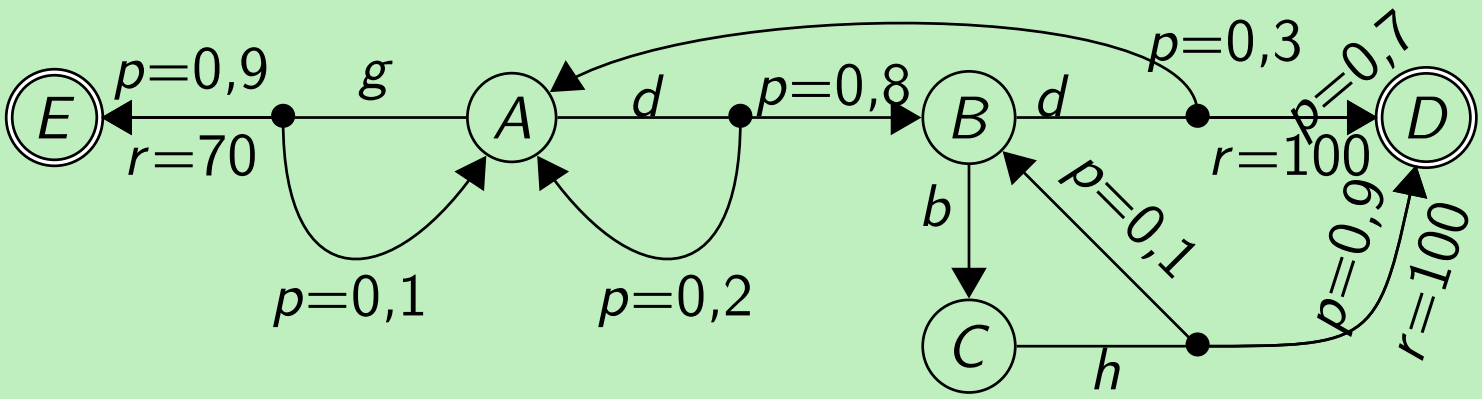
Exemple :



- séquence A, g, E : $R = 70$, $p = 0,9$
- séquence A, g, A, g, E : $R = 70\gamma$, $p = 0,1 \times 0,9 = 0,09$

MDP : Cas non déterministe

Exemple :



- séquence A, g, E : $R = 70$, $p = 0,9$
- séquence A, g, A, g, E : $R = 70\gamma$, $p = 0,1 \times 0,9 = 0,09$
- séquence A, g, A, g, A, g, E : $R = 70\gamma^2$, $p = 0,1^2 \times 0,9 = 0,009$

MDP : Cas non déterministe

Modèle :

$p(s, a, s_1, r) =$ probabilité d'arriver en s_1 avec la récompense r, en réalisation l'action a dans l'état s

MDP : Cas non déterministe

Modèle :

$p(s, a, s_1, r) =$ probabilité d'arriver en s_1 avec la récompense r,
en réalisation l'action a dans l'état s

Qualité d'une action : $Q^*(s, a) =$ espérance de récompense accumulée

- sur toutes les trajectoires possibles commençant par s, a
- en supposant qu'on choisit à chaque pas / état s_i
  l'action a_i qui est *la meilleure en moyenne*

MDP : Cas non déterministe

Modèle :

$p(s, a, s_1, r) =$ probabilité d'arriver en s_1 avec la récompense r,
en réalisation l'action a dans l'état s

Qualité d'une action : $Q^*(s, a) =$ espérance de récompense accumulée

- sur toutes les trajectoires possibles commençant par s, a
- en supposant qu'on choisit à chaque pas / état s_i
  l'action a_i qui est *la meilleure en moyenne*

Récompense optimale : on peut aussi définir $R^*(s) = \max_{a \in \mathcal{A}(s)} Q^*(s, a)$

MDP : Cas non déterministe

Propriété (Équations de Bellman)

Pour tous $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$$Q^*(s,a) = \sum_{s_1,r} p(s,a,s_1,r)\big[r + \gamma \max_{a_1 \in \mathcal{A}(s_1)} Q^*(s_1,a_1)\big]$$

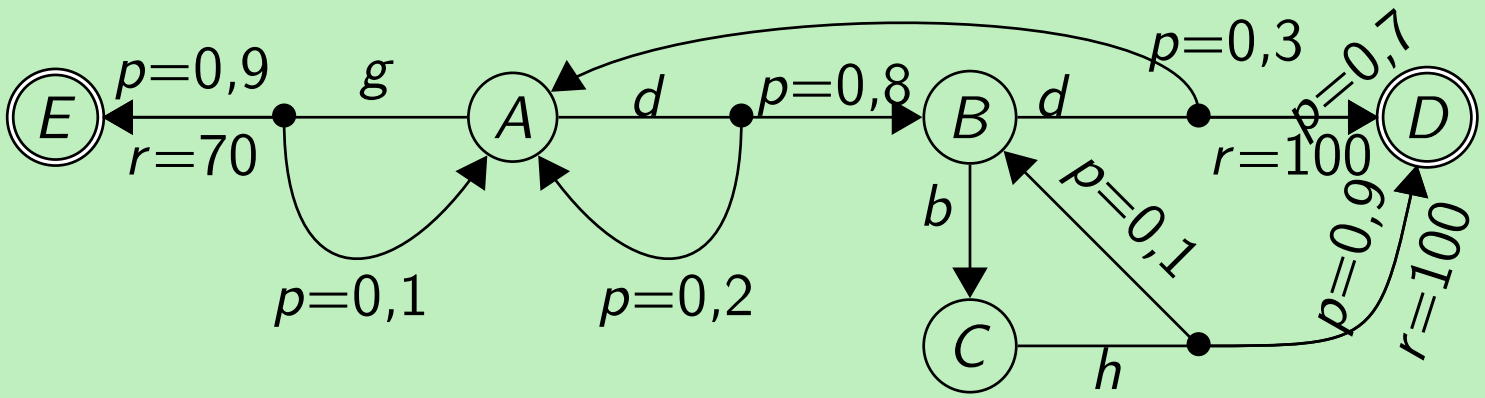
MDP : Cas non déterministe

Propriété (Équations de Bellman)

Pour tous $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$$Q^*(s,a) = \sum_{s_1,r} p(s,a,s_1,r)\big[r + \gamma \max_{a_1 \in \mathcal{A}(s_1)} Q^*(s_1,a_1)\big]$$

Exemple :



- $Q^*(D)=Q^*(E)=0$ *(états terminaux)*

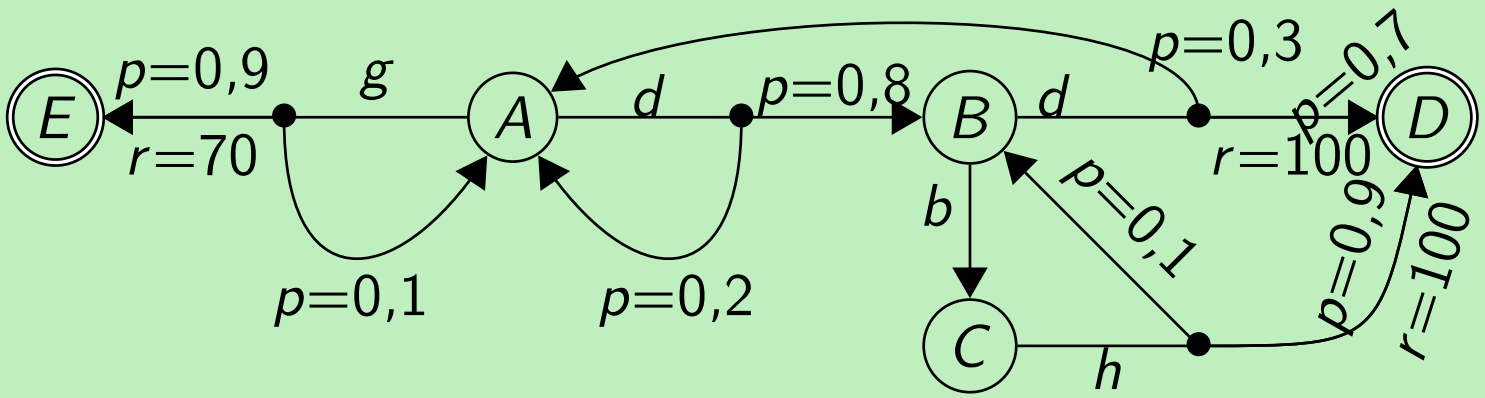
MDP : Cas non déterministe

Propriété (Équations de Bellman)

Pour tous $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$$Q^*(s,a) = \sum_{s_1,r} p(s,a,s_1,r)\big[r + \gamma \max_{a_1 \in \mathcal{A}(s_1)} Q^*(s_1,a_1)\big]$$

Exemple :



- $Q^*(D)=Q^*(E)=0$ *(états terminaux)*
- $Q^*(C,h)=0{,}1\big[0+\gamma\max\{Q^*(B,b),Q^*(B,d)\}\big]+0{,}9\big[100+\gamma\max\{Q^*(D)\}\big]$

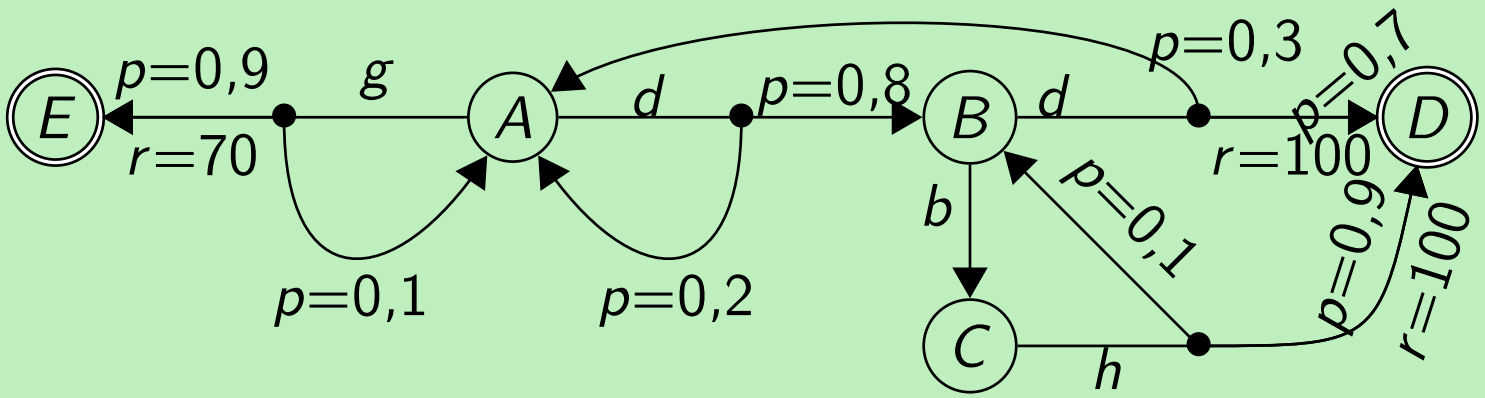
MDP : Cas non déterministe

Propriété (Équations de Bellman)

Pour tous $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$$Q^*(s,a) = \sum_{s_1,r} p(s,a,s_1,r)\big[r + \gamma \max_{a_1 \in \mathcal{A}(s_1)} Q^*(s_1,a_1)\big]$$

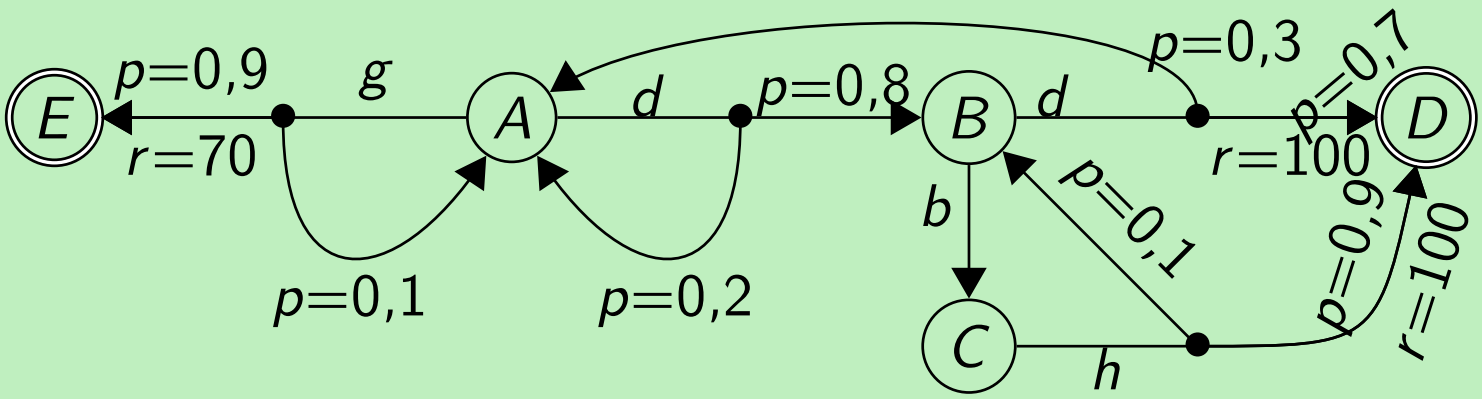
Exemple :



- $Q^*(D)=Q^*(E)=0$ *(états terminaux)*
- $Q^*(C,h)=0{,}1\big[0+\gamma\max\{Q^*(B,b),Q^*(B,d)\}\big]+0{,}9\big[100+\gamma\max\{Q^*(D)\}\big]$
- $Q^*(B,b)=\gamma Q^*(C,h)$

MDP : Cas non déterministe

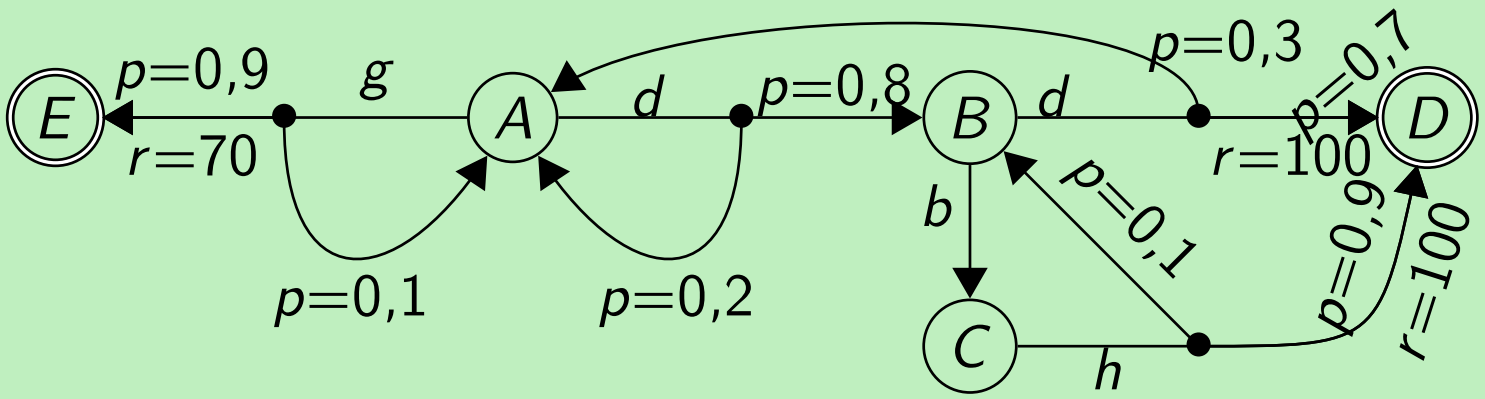
Exemple :



- $Q^*(D)=Q^*(E)=0$ *(états terminaux)*
- $Q^*(C,h)=0{,}1\big[0+\gamma\max\{Q^*(B,b),Q^*(B,d)\}\big]+0{,}9\big[100+\gamma\max\{Q^*(D)\}\big]$
- $Q^*(B,b)=\gamma Q^*(C,h)$
- $Q^*(B,d)=0{,}3\big[0+\gamma\max\{Q^*(A,g),Q^*(A,d)\}\big]$

MDP : Cas non déterministe

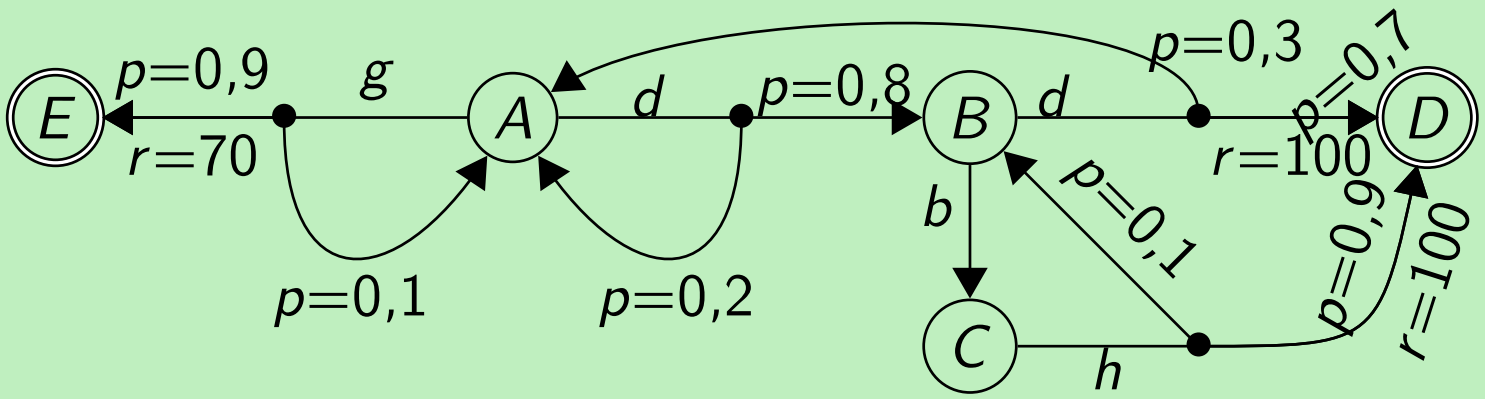
Exemple :



- $Q^*(D)=Q^*(E)=0$ *(états terminaux)*
- $Q^*(C,h)=0{,}1\big[0+\gamma\max\{Q^*(B,b),Q^*(B,d)\}\big]+0{,}9\big[100+\gamma\max\{Q^*(D)\}\big]$
- $Q^*(B,b)=\gamma Q^*(C,h)$
- $Q^*(B,d)=0{,}3\big[0+\gamma\max\{Q^*(A,g),Q^*(A,d)\}\big]$
- $Q^*(A,g)=0{,}9\big[70+\gamma\max\{Q^*(E)\}\big]+0{,}1\big[0+\gamma\max\{Q^*(A,g),Q^*(A,d)\}\big]$

MDP : Cas non déterministe

Exemple :



- $Q^*(D)=Q^*(E)=0$ *(états terminaux)*
- $Q^*(C,h)=0{,}1\big[0+\gamma\max\{Q^*(B,b),Q^*(B,d)\}\big]+0{,}9\big[100+\gamma\max\{Q^*(D)\}\big]$
- $Q^*(B,b)=\gamma Q^*(C,h)$
- $Q^*(B,d)=0{,}3\big[0+\gamma\max\{Q^*(A,g),Q^*(A,d)\}\big]$
- $Q^*(A,g)=0{,}9\big[70+\gamma\max\{Q^*(E)\}\big]+0{,}1\big[0+\gamma\max\{Q^*(A,g),Q^*(A,d)\}\big]$
- $Q^*(A,d)=0{,}8\big[0+\gamma\max\{Q^*(B,d),Q^*(B,b)\}\big]$
  $+0{,}2\big[0+\gamma\max\{Q^*(A,g),Q^*(A,d)\}\big]$

MDP : Cas non déterministe

Propriété (Équations de Bellman)

Pour tous $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$$Q^*(s,a) = \sum_{s_1,r} p(s,a,s_1,r)\big[r + \gamma \max_{a_1 \in \mathcal{A}(s_1)} Q^*(s_1,a_1)\big]$$

- Équations ni plus ni moins linéaires que dans le cas déterministe !
- Mêmes méthodes de résolution

Programmation dynamique

Prorammation dynamique pour MDP

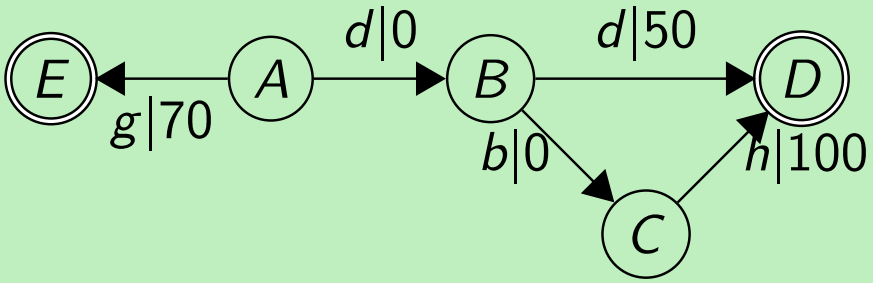
1. $R(s) \leftarrow 0$ pour tout $s \in \mathcal{S}$;
2. Répéter :
   a. pour tout $s \in \mathcal{S}$, s non terminal, faire :
      i. pour tout $a \in \mathcal{A}(s)$: $Q(s,a) \leftarrow \sum_{s_1,r} p(s,a,s_1,r)(r + \gamma R(s_1))$
      ii. $R(s) \leftarrow \max_{a \in \mathcal{A}(s)} Q(s,a)$;

Programmation dynamique

Prorammation dynamique pour MDP

1. $R(s) \leftarrow 0$ pour tout $s \in \mathcal{S}$;
2. Répéter :
   a. pour tout $s \in \mathcal{S}$, s non terminal, faire :
      i. pour tout $a \in \mathcal{A}(s)$: $Q(s,a) \leftarrow \sum_{s_1,r} p(s,a,s_1,r)(r + \gamma R(s_1))$
      ii. $R(s) \leftarrow \max_{a \in \mathcal{A}(s)} Q(s,a)$;

Exemple : Cas déterministe



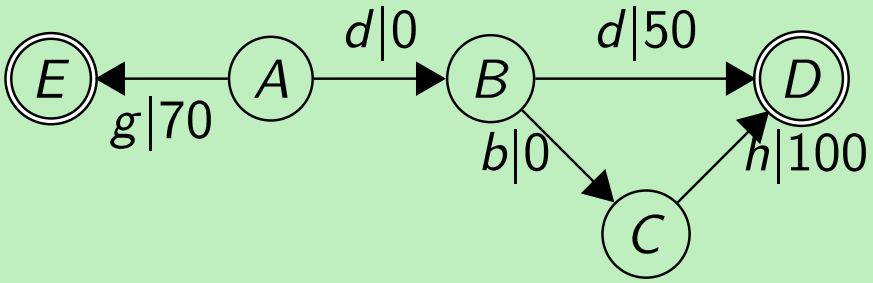
γ=0,9	E	A,g	A,d	B,d	B,b	C,h	D
init :R	0	0		0		0	0

Programmation dynamique

Prorammation dynamique pour MDP

1. $R(s) \leftarrow 0$ pour tout $s \in \mathcal{S}$;
2. Répéter :
   a. pour tout $s \in \mathcal{S}$, s non terminal, faire :
      i. pour tout $a \in \mathcal{A}(s)$: $Q(s,a) \leftarrow \sum_{s_1,r} p(s,a,s_1,r)(r + \gamma R(s_1))$
      ii. $R(s) \leftarrow \max_{a \in \mathcal{A}(s)} Q(s,a)$;

Exemple : Cas déterministe



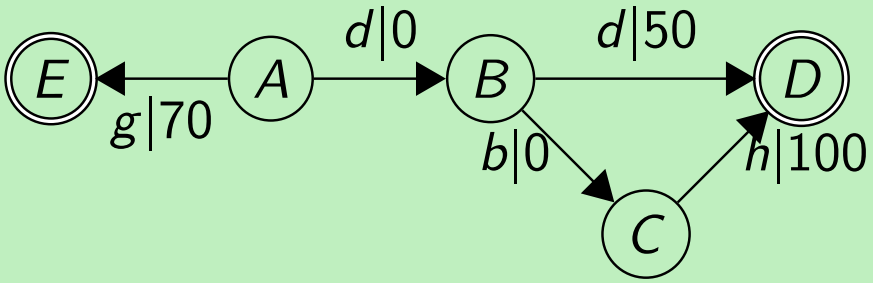
γ=0,9	E	A,g	A,d	B,d	B,b	C,h	D
init :R	0	0		0		0	0
it. 1, Q :	0	$70+\gamma\times0$	$0+\gamma\times0$	$50+\gamma\times0$	$0+\gamma\times0$	$100+\gamma\times0$	0

Programmation dynamique

Prorammation dynamique pour MDP

1. $R(s) \leftarrow 0$ pour tout $s \in \mathcal{S}$;
2. Répéter :
   a. pour tout $s \in \mathcal{S}$, s non terminal, faire :
      i. pour tout $a \in \mathcal{A}(s)$: $Q(s,a) \leftarrow \sum_{s_1,r} p(s,a,s_1,r)(r + \gamma R(s_1))$
      ii. $R(s) \leftarrow \max_{a \in \mathcal{A}(s)} Q(s,a)$;

Exemple : Cas déterministe



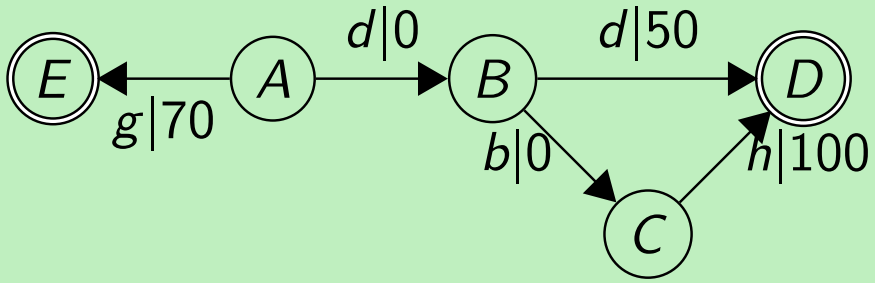
γ=0,9	E	A,g	A,d	B,d	B,b	C,h	D
init :R	0	0		0		0	0
it. 1, Q :	0	70	0	50	0	0	0

Programmation dynamique

Prorammation dynamique pour MDP

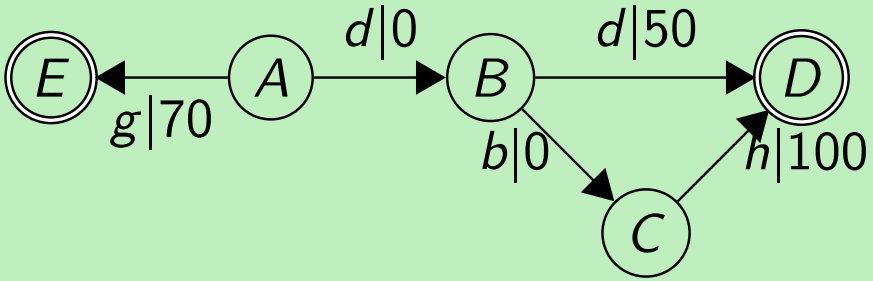
1. $R(s) \leftarrow 0$ pour tout $s \in \mathcal{S}$;
2. Répéter :
   a. pour tout $s \in \mathcal{S}$, s non terminal, faire :
      i. pour tout $a \in \mathcal{A}(s)$: $Q(s,a) \leftarrow \sum_{s_1,r} p(s,a,s_1,r)(r + \gamma R(s_1))$
      ii. $R(s) \leftarrow \max_{a \in \mathcal{A}(s)} Q(s,a)$;

Exemple : Cas déterministe



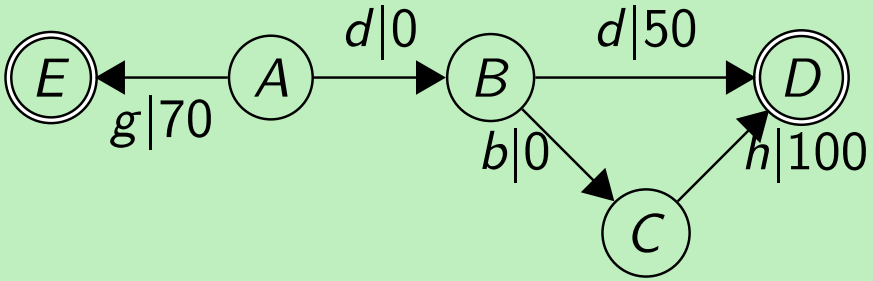
γ=0,9	E	A,g	A,d	B,d	B,b	C,h	D
init :R	0	0		0		0	0
it. 1, Q :	0	70	0	50	0	0	0
it. 1, R :	0	70		50		100	0

Exemple : Cas déterministe



γ=0,9	E	A,g	A,d	B,d	B,b	C,h	D
init :R	0	0		0		0	0
it. 1, Q :	0	70	0	50	0	0	0
it. 1, R :	0	70		50		100	0
it. 2, Q :	0	70	$0+50\gamma$ =45	50	$0+100\gamma$ =90	100	0
it. 2, R :	0	70		90		100	0

Exemple : Cas déterministe



γ=0,9	E	A,g	A,d	B,d	B,b	C,h	D
init :R	0	0		0		0	0
it. 1, Q :	0	70	0	50	0	0	0
it. 1, R :	0	70		50		100	0
it. 2, Q :	0	70	$0+50\gamma$ =45	50	$0+100\gamma$ =90	100	0
it. 2, R :	0	70		90		100	0
it. 3, Q :	0	70	$0+90\gamma$	50	90	100	0
it. 3, R :	0	81		90		100	0

Programmation dynamique

Prorammation dynamique pour MDP

1. $R(s) \leftarrow 0$ pour tout $s \in \mathcal{S}$;
2. Répéter :
   a. pour tout $s \in \mathcal{S}$, s non terminal, faire :
      i. pour tout $a \in \mathcal{A}(s)$: $Q(s,a) \leftarrow \sum_{s_1,r} p(s,a,s_1,r)(r + \gamma R(s_1))$
      ii. $R(s) \leftarrow \max_{a \in \mathcal{A}(s)} Q(s,a)$;

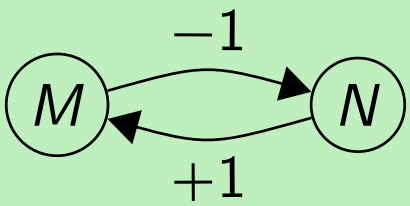
L'algorithme converge vers R^*, Q^* *à la limite*.

Prorammation dynamique pour MDP

1. $R(s) \leftarrow 0$ pour tout $s \in \mathcal{S}$;
2. Répéter :
   a. pour tout $s \in \mathcal{S}$, s non terminal, faire :
      i. pour tout $a \in \mathcal{A}(s)$: $Q(s,a) \leftarrow \sum_{s_1,r} p(s,a,s_1,r)(r + \gamma R(s_1))$
      ii. $R(s) \leftarrow \max_{a \in \mathcal{A}(s)} Q(s,a)$;

L'algorithme converge vers R^*, Q^* *à la limite*.

Exemple :



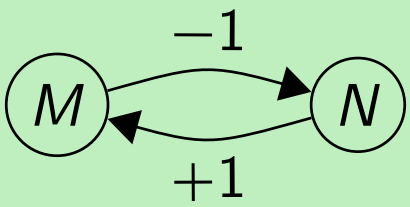
Programmation dynamique

Prorammation dynamique pour MDP

1. $R(s) \leftarrow 0$ pour tout $s \in \mathcal{S}$;
2. Répéter :
   a. pour tout $s \in \mathcal{S}$, s non terminal, faire :
      i. pour tout $a \in \mathcal{A}(s)$: $Q(s,a) \leftarrow \sum_{s_1,r} p(s,a,s_1,r)(r + \gamma R(s_1))$
      ii. $R(s) \leftarrow \max_{a \in \mathcal{A}(s)} Q(s,a)$;

L'algorithme converge vers R^*, Q^* *à la limite*.

Exemple :



Conditions d'arrêt – surtout s'il y a des cycles :

- plus de temps
- peu d'écart entre deux itérations successives

Apprentissage par renforcement

Lorsqu'on connaît le MDP / les proba. de transitions / les récompenses :

- *résoudre* le MDP = calculer la fonctions Q^* associée
- programmation dynamique possible

Apprentissage par renforcement

Lorsqu'on connaît le MDP / les proba. de transitions / les récompenses :

- *résoudre* le MDP = calculer la fonctions Q^* associée
- programmation dynamique possible

Comment *résoudre* un MDP qu'on ne connaît pas à l'avance ?

Apprentissage par renforcement

Lorsqu'on connaît le MDP / les proba. de transitions / les récompenses :

- *résoudre* le MDP = calculer la fonctions Q^* associée
- programmation dynamique possible

Comment *résoudre* un MDP qu'on ne connaît pas à l'avance ?

En apprentissage par renforcement :

- l'environnement est (partiellement) inconnu
- l'environnement peut changer
- on peut interagir avec l'environnement

Apprentissage par renforcement

Lorsqu'on connaît le MDP / les proba. de transitions / les récompenses :

- *résoudre* le MDP = calculer la fonctions Q^* associée
- programmation dynamique possible

Comment *résoudre* un MDP qu'on ne connaît pas à l'avance ?

En apprentissage par renforcement :

- l'environnement est (partiellement) inconnu
- l'environnement peut changer
- on peut interagir avec l'environnement

Il va falloir ***explorer*** le MDP, l'échantillonner.
On ne suppose plus qu'on connaît les fonctions r et δ.

Apprentissage par renforcement : Algorithme générique

Algorithme générique d'apprentissage par renforcement

1. initialiser une représentation du MDP
2. répéter
   a. choisir un état de départ $s \in \mathcal{S}$;
   b. répéter
      i. choisir une action $a \in \mathcal{A}(s)$;
      ii. exécuter a en observant récompense r + nouvel état s' ;
      iii. mettre à jour la représentation du MDP ;
      iv. $s \leftarrow s'$.

où

- représentation du MDP = fonctions R, ou Q, ou ...
- choix en 2.c.ii : stratégie bornée, ϵ-gourmande, ...
- mise à jour en 2.c.iii : coef. α d'apprentissage, ...

Apprentissage par renforcement : Q-learning

Algorithme Q-learning

1. $Q(s, a) \leftarrow 0$ pour tout s, $a \in \mathcal{A}(s)$;
2. répéter
   a. choisir un état de départ $s \in \mathcal{S}$;
   b. répéter
      i. choisir une action $a \in \mathcal{A}(s)$, stratégie ϵ-gourmande ;
      ii. exécuter a en observant récompense r + nouvel état s' ;
      iii. $Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max_{a' \in \mathcal{A}(s')} Q(s', a'))$;
      iv. $s \leftarrow s'$.

- Représente le MDP avec la fonction Q, initialisée à 0
- Stratégie d'exploration ϵ-gourmande
- Règle de mise à jour :

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max_{a' \in \mathcal{A}(s')} Q(s', a'))$$

$\alpha \in [0, 1] =$ ***taux d'apprentissage*** par ex. : $\alpha = 1/(1 + \text{nb_visites}(s, a))$

Apprentissage par renforcement : Q-learning

Propriété

Convergence de l'algorithme de Q-learning :

- Si les récompenses immédiates sont bornées, et
- si $0 \leq \gamma < 1$, et
- si le choix de l'action à effectuer dans la boucle est tel que toute paire état/action sera évaluée infiniment souvent

alors la fonction Q apprise par l'algorithme converge vers la fonction $Q(\pi^*, \cdot, \cdot)$ d'une politique π^* optimale.

Apprentissage par renforcement : Q-learning

Propriété

Convergence de l'algorithme de Q-learning :

- Si les récompenses immédiates sont bornées, et
- si $0 \leq \gamma < 1$, et
- si le choix de l'action à effectuer dans la boucle est tel que toute paire état/action sera évaluée infiniment souvent

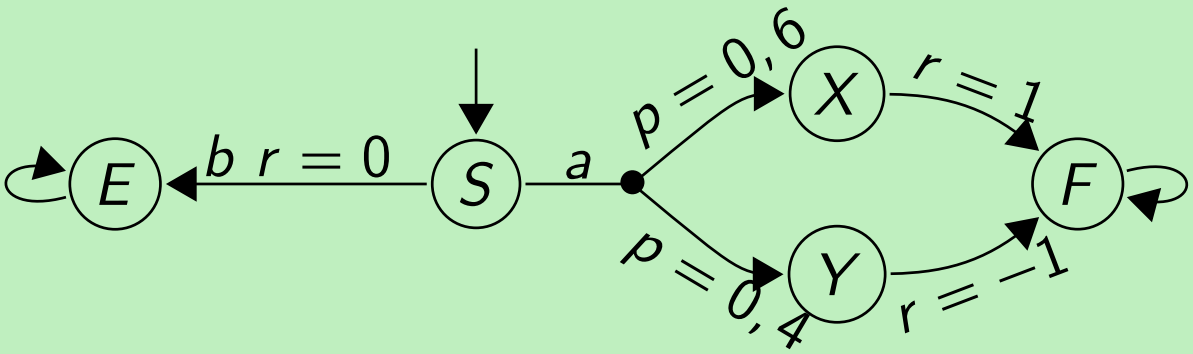
alors la fonction Q apprise par l'algorithme converge vers la fonction $Q(\pi^*, \cdot, \cdot)$ d'une politique π^* optimale.

Amélioration possible : lorsque les récompenses apparaissent peu souvent, on peut rétropropager Q pour toutes les paires état/action rencontrées lors d'un épisode.

Apprentissage par renforcement : Q-learning

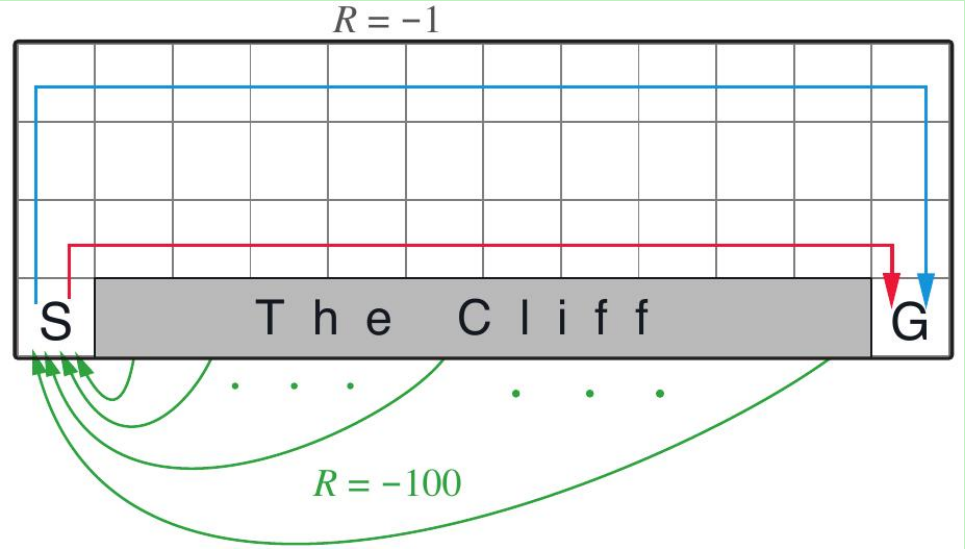
Avec l'algorithme de Q-learning : risque d'oscillations

Exemple :



Apprentissage par renforcement : SARSA

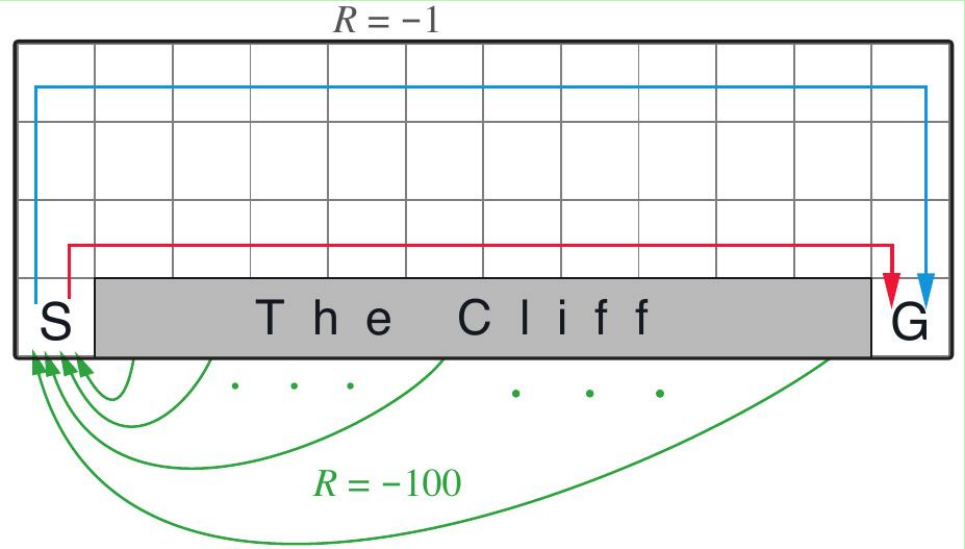
Exemple :



- La politique optimale (rouge) passe le long de la falaise.

Apprentissage par renforcement : SARSA

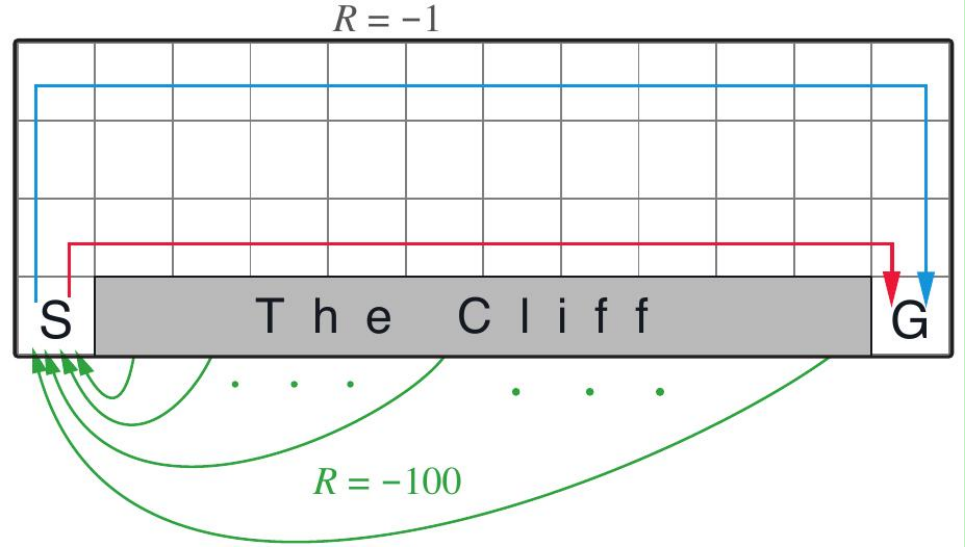
Exemple :

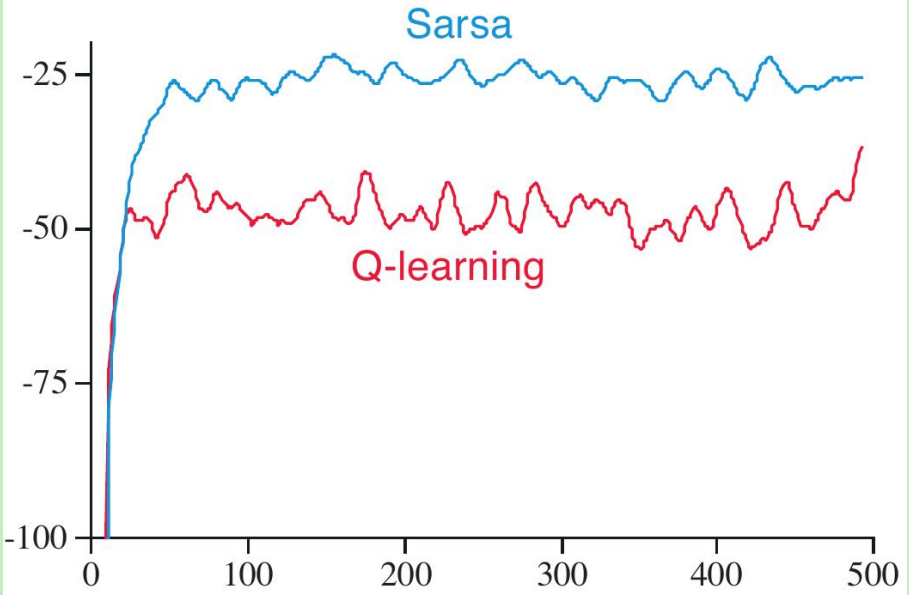


- La politique optimale (rouge) passe le long de la falaise.
- Mais exploration : de temps en temps (avec proba ϵ) l'agent saute ! !

Apprentissage par renforcement : SARSA

Exemple :





récompenses accumulées / nb. itérations

(Figures Sutton and Barto [2018])

- La politique optimale (rouge) passe le long de la falaise.
- Mais exploration : de temps en temps (avec proba ϵ) l'agent saute ! !

$\Rightarrow$ Lourdes pertes durant la phase d'apprentissage
(tant que ϵ pas trop petit)

Apprentissage par renforcement : SARSA

SARSA

1. $Q(s, a) \leftarrow 0$ pour tout s, $a \in \mathcal{A}(s)$;
2. répéter
   a. choisir un état de départ $s \in \mathcal{S}$;
   b. choisir une action $a \in \mathcal{A}(s)$, stratégie ϵ-gourmande ;
   c. répéter
      i. exécuter a en observant récompense r + nouvel état s' ;
      ii. choisir une action $a' \in \mathcal{A}(s')$, stratégie ϵ-gourmande ;
      iii. $Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(r + \gamma Q(s', a'))$;
      iv. $s \leftarrow s'$, $a \leftarrow a'$.

Apprentissage par renforcement : SARSA

SARSA

1. $Q(s, a) \leftarrow 0$ pour tout s, $a \in \mathcal{A}(s)$;
2. répéter
   a. choisir un état de départ $s \in \mathcal{S}$;
   b. choisir une action $a \in \mathcal{A}(s)$, stratégie ϵ-gourmande ;
   c. répéter
      i. exécuter a en observant récompense r + nouvel état s' ;
      ii. choisir une action $a' \in \mathcal{A}(s')$, stratégie ϵ-gourmande ;
      iii. $Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(r + \gamma Q(s', a'))$;
      iv. $s \leftarrow s'$, $a \leftarrow a'$.

- Algorithme «*on policy*» : on met à jour avec la qualité de l'action correspondant à l'action déterminée par la politique
- Avec cet algorithme, à ϵ constant :
  l'algorithme converge vers un chemin plus éloigné de la falaise.
- Si ϵ tends vers 0, les deux algorithmes tendent vers la politique optimale.

Apprentissage par renforcement : Généralisation des renforcements

- Espace d'états souvent gigantesque ($|\mathcal{S}| \times E[|\mathcal{A}(s)|] \sim 10^{29}$ pour Othello !)

Apprentissage par renforcement : Généralisation des renforcements

- Espace d'états souvent gigantesque ($|\mathcal{S}| \times E[|\mathcal{A}(s)|] \sim 10^{29}$ pour Othello !)
- Impossible de mémoriser une «*table*» pour Q

Apprentissage par renforcement : Généralisation des renforcements

- Espace d'états souvent gigantesque ($|\mathcal{S}| \times E[|\mathcal{A}(s)|] \sim 10^{29}$ pour Othello !)
- Impossible de mémoriser une «*table*» pour Q
- Une telle table contiendrait de très nombreuses cases non évaluées

Apprentissage par renforcement : Généralisation des renforcements

- Espace d'états souvent gigantesque ($|\mathcal{S}| \times E[|\mathcal{A}(s)|] \sim 10^{29}$ pour Othello !)
- Impossible de mémoriser une «*table*» pour Q
- Une telle table contiendrait de très nombreuses cases non évaluées

$\Rightarrow$ Apprendre une fonction Q dans un espace de dim. contrôlée

Apprentissage par renforcement : Généralisation des renforcements

- Espace d'états souvent gigantesque ($|\mathcal{S}| \times E[|\mathcal{A}(s)|] \sim 10^{29}$ pour Othello !)
- Impossible de mémoriser une «*table*» pour Q
- Une telle table contiendrait de très nombreuses cases non évaluées

$\Rightarrow$ Apprendre une fonction Q dans un espace de dim. contrôlée

$\Rightarrow$ Par ex. : apprendre un réseau de neuronnes

Apprentissage par renforcement : Généralisation des renforcements

- Espace d'états souvent gigantesque ($|\mathcal{S}| \times E[|\mathcal{A}(s)|] \sim 10^{29}$ pour Othello !)
- Impossible de mémoriser une «*table*» pour Q
- Une telle table contiendrait de très nombreuses cases non évaluées

$\Rightarrow$ Apprendre une fonction Q dans un espace de dim. contrôlée

$\Rightarrow$ Par ex. : apprendre un réseau de neuronnes

Apprentissage par renforcement : Généralisation des renforcements

- Espace d'états souvent gigantesque ($|\mathcal{S}| \times E[|\mathcal{A}(s)|] \sim 10^{29}$ pour Othello !)
- Impossible de mémoriser une «*table*» pour Q
- Une telle table contiendrait de très nombreuses cases non évaluées

$\Rightarrow$ Apprendre une fonction Q dans un espace de dim. contrôlée

$\Rightarrow$ Par ex. : apprendre un réseau de neuronnes

Par exemple, pour Othello :

- 64 unités en entrées
- une couche cachée
- 64 unités en sorties : $Q(s, a)$ pour chacun des coups possibles a

Modification des algorithmes

À chaque fois qu'on doit mettre Q à jour : rétropropagation.

Apprentissage par renforcement : Généralisation des renforcements

- Espace d'états souvent gigantesque ($|\mathcal{S}| \times E[|\mathcal{A}(s)|] \sim 10^{29}$ pour Othello !)
- Impossible de mémoriser une «*table*» pour Q
- Une telle table contiendrait de très nombreuses cases non évaluées

$\Rightarrow$ Apprendre une fonction Q dans un espace de dim. contrôlée

$\Rightarrow$ Par ex. : apprendre un réseau de neuronnes

Par exemple, pour Othello :

- 64 unités en entrées
- une couche cachée
- 64 unités en sorties : $Q(s, a)$ pour chacun des coups possibles a

Modification des algorithmes

À chaque fois qu'on doit mettre Q à jour : rétropropagation.

On apprend une approximation de Q

$\Rightarrow$ aucune garantie de convergence (mais marche bien en pratique)

Apprentissage par renforcement : Généralisation des renforcements

Q-learning avec généralisation

1. Initialiser Q ;
2. répéter
   a. choisir $s \in \mathcal{S}$; $\mathcal{E} \leftarrow \emptyset$;
   b. répéter :
      i. choisir $a \in \mathcal{A}(s)$;
      ii. $r \leftarrow r(s,a)$; $s' \leftarrow \delta(s,a)$; $\alpha \leftarrow 1/(1 + \text{nb_visites}(s,a))$;
      iii. $q \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a' \in \mathcal{A}(s)} Q(s',a'))$;
      iv. $\mathcal{E} \leftarrow \mathcal{E} \cup \{((s,a),q)\}$;
      v. $s \leftarrow s'$;
   c. « *ré-apprendre* » Q à partir des exemples de $\mathcal{E}$;

En 2.c : ***régression***.

Apprentissage par renforcement : Généralisation des renforcements

Q-learning avec généralisation

1. Initialiser Q ;
2. répéter
   a. choisir $s \in \mathcal{S}$; $\mathcal{E} \leftarrow \emptyset$;
   b. répéter :
      i. choisir $a \in \mathcal{A}(s)$;
      ii. $r \leftarrow r(s,a)$; $s' \leftarrow \delta(s,a)$; $\alpha \leftarrow 1/(1 + \text{nb_visites}(s,a))$;
      iii. $q \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a' \in \mathcal{A}(s)} Q(s',a'))$;
      iv. $\mathcal{E} \leftarrow \mathcal{E} \cup \{((s,a),q)\}$;
      v. $s \leftarrow s'$;
   c. « *ré-apprendre* » Q à partir des exemples de $\mathcal{E}$;

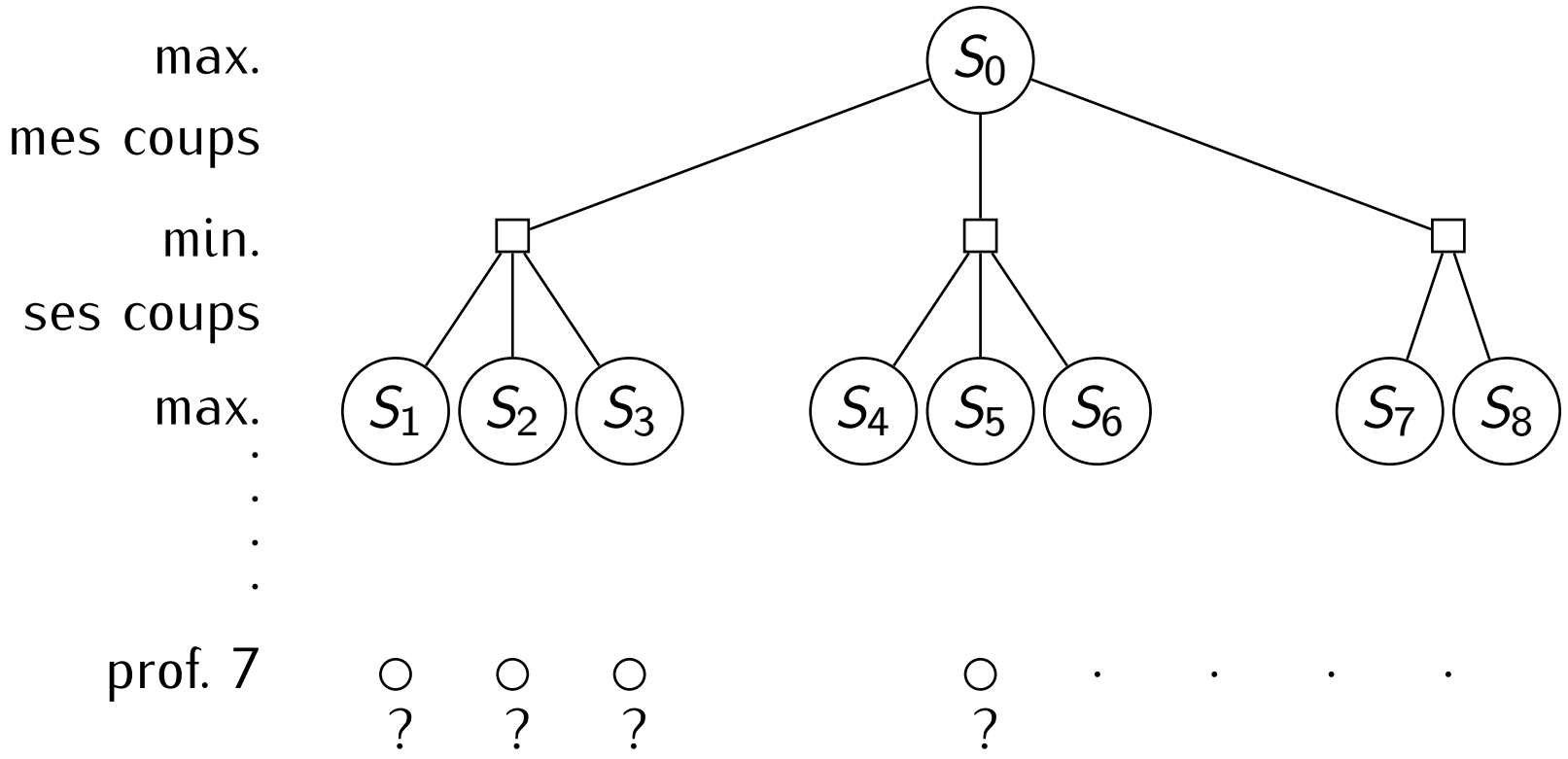
En 2.c : ***régression***.

Problème : dès lors qu'on a une telle représentation appochée de Q, la méthode risque de ne pas converger.

Apprentissage par renforcement : Arbres de jeux

Évaluation heuristique :

- Arbre de recherche à horizon fixé
- Quand on arrive à la prof. max. : fonction d'évaluation de la position

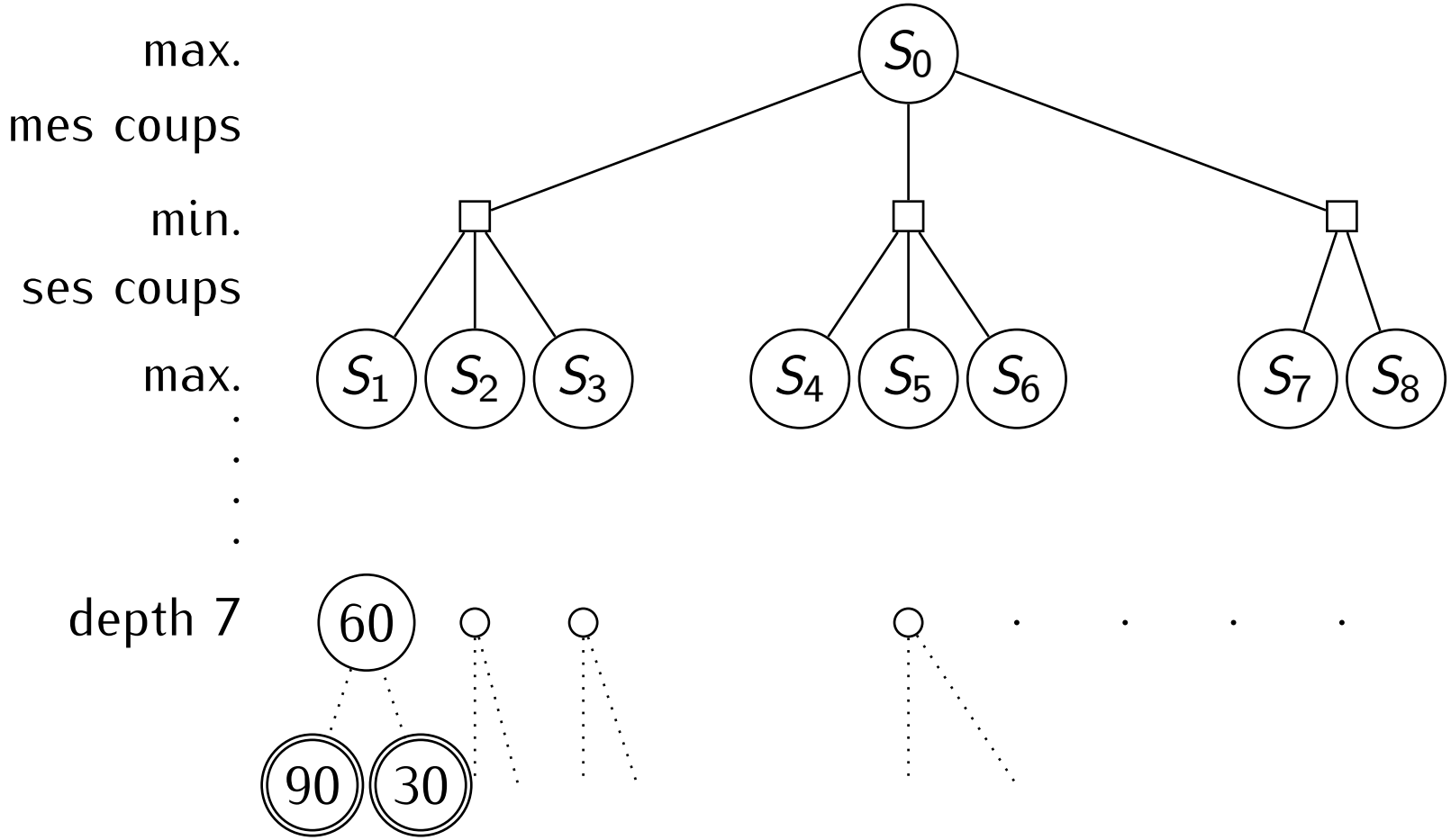


Marche bien pour jeux à 2, somme nulle, si bonne heuristique

DeepBlue a battu Kasparov aux échecs en 1997

Apprentissage par renforcement : Arbres de jeux

Évaluation «*Monte-Carlo*» : à chaque fois qu'on a besoin de la valeur d'un nœud : simuler k fin de parties aléatoires



Apprentissage par renforcement : Arbres de jeux

Évaluation «*Monte-Carlo*» : à chaque fois qu'on a besoin de la valeur d'un nœud : simuler k fin de parties aléatoires

- Pratique quand on n'a pas de bonne heuristique (Go, «*General Game Playing*»)
- Méthode optimiste : suppose que l'adversaire joue au hasard
- CadiaPlayer a gagné trois années de suite la compétition de GGP dans les années 2000

Monte-Carlo Tree Search (MCTS)
Proposé pour le jeu de Go dans les années 2000

Monte-Carlo Tree Search (MCTS)

Proposé pour le jeu de Go dans les années 2000

- pas d'horizon fixé
- on peut perdre moins de temps à évaluer les séquences peu prometteuses
- on utilise l'information / les évaluations dès que trouvées
- on ne développe que les branches prometteuses

$\Rightarrow$ on peut descendre plus profondément dans ces branches

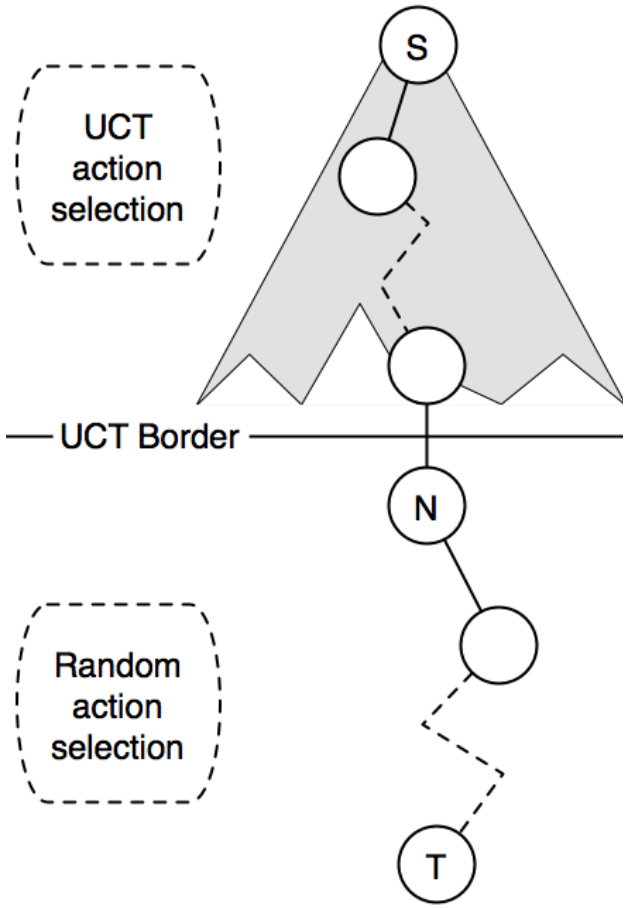
Apprentissage par renforcement : Arbres de jeux

Arbre de recherche étendu par épisodes :

- on part de la racine
- dans la partie déjà visitée :
  sélection d'un successeur selon UCT
- au premier nœud par encore rencontré :
  - on ajoute les successeurs
  - simul. → une fin de partie
  - rétropropagation

Pour la sélection à l'état s :
sélectionner l'action a qui maximise

$$Q(s,a) + C\sqrt{\frac{\log N(s)}{N(s,a)}}$$



(fig. from CadiaPlayer - Bjornsson and Finsson, 2009)

Pour aller plus loin

- Sujet vaste ! !
- Plein d'algorithmes différents
- Pas facile à mettre en œuvre – beaucoup d'*engineering*
- Du 25 mars au 9 avril 2021 :
  ANITI : École virtuelle – reinforcement learning
  Ouverte à tou-te-s et gratuite
  `https://rlvs.aniti.fr/`

David A. Cohn, Les E. Atlas, and Richard E. Ladner. Improving generalization with active learning. ***Machine Learning***, 15(2) :201–221, 1994. URL `http://dx.doi.org/10.1007/BF00993277`.

Sanjoy Dasgupta. Two faces of active learning. ***Theoretical Computer Science***, 412(19) :1767–1781, 2011.

Shai Shalev-Shwartz and Shai Ben-David. ***Understanding machine learning : From theory to algorithms***. Cambridge university press, 2014.

Richard S Sutton and Andrew G Barto. ***Reinforcement Learning : An Introduction***. The MIT Press, 2nd edition, 2018. URL `http://incompleteideas.net/book/the-book.html`.