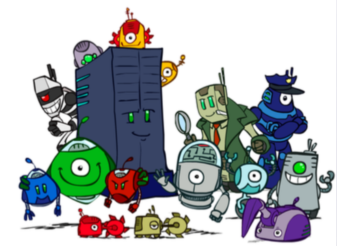
Modèles pour la prise de décision sous incertitudes & Applications robotiques

Laëtitia Matignon
laetitia.matignon@univ-lyon1.fr

Equipe SMA@LIRIS

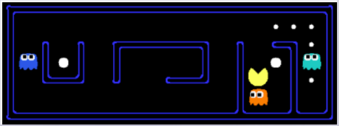


LIRIS
Lyon 1

Organisation des séances

- CM1 + TP1 : Planification sous incertitudes
- CM2 + TP2 : Apprentissage par renforcement

Objectifs : savoir modéliser un problème sous forme de processus décisionnel markovien (MDP), savoir résoudre le MDP lorsque le modèle est connu (planification) ou partiellement connu (apprentissage).



Objectif des TPs : implémenter un agent qui **apprend** à jouer à Pacman

Evaluation : une note sur l'ensemble des TPs, une partie au CF.

Planification sous incertitudes - Processus décisionnel markovien

Laëtitia Matignon

LIRIS

Lyon 1

Plan

Intelligence artificielle et agent ?

VS

VS

Intelligence artificielle et agent ?

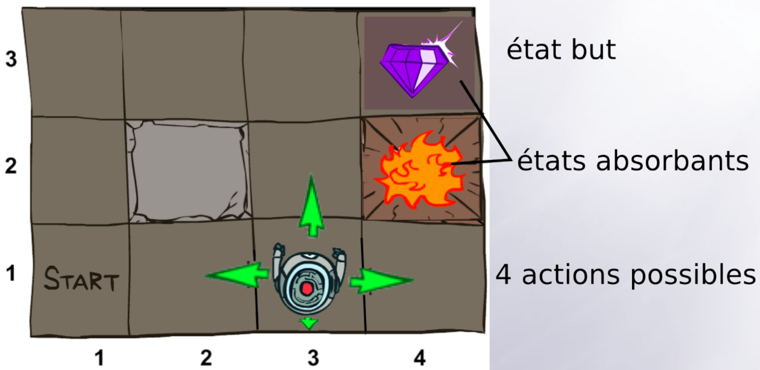
VS

VS

Agent = entité autonome évoluant dans un environnement

- Capteurs : *percevoir l'environnement*
- Actionneurs : *agir sur l'environnement*
- Objectif : *prendre des décisions pour atteindre son objectif*

Problème posé



Perceptions du robot/agent : sa case du labyrinthe.

Comment doit agir le robot pour atteindre le plus rapidement possible l'état objectif ?

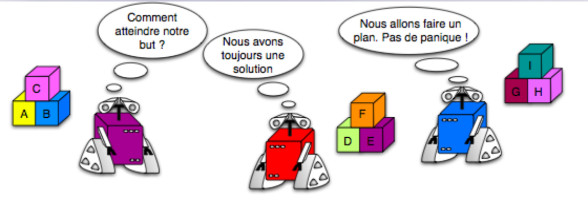
Problème de planification

Définition

Planning is the reasoning side of acting. It is an abstract, explicit deliberation process that chooses and organizes actions by anticipating their expected outcomes. This deliberation aims at achieving as best as possible some pre-stated objectives. *[Automated Planning, M. Ghallab et al. Morgan Kaufmann, 2004]*

Planifier

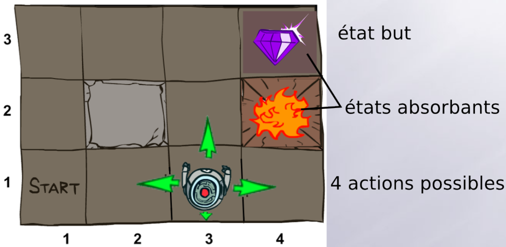
Trouver un plan pour aller d'un état initial à un état but en respectant certains objectifs. Un plan est une séquence d'actions.



Problème de planification

Planifier

Trouver un plan ou une séquence d'actions pour aller d'un état initial à un état but en respectant certains objectifs.



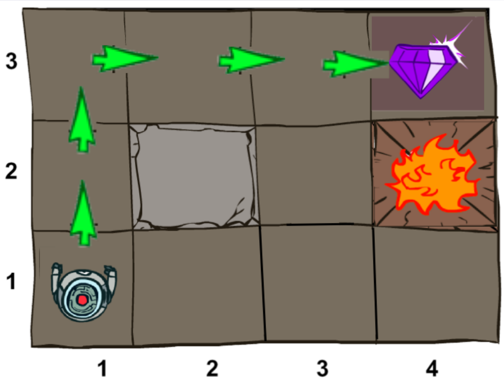
Perceptions de l'agent : sa position (case) dans le labyrinthe.

Comment trouver la plus petite séquence d'actions menant au but depuis l'état initial ?

Problème de planification

Algorithme de recherche

A* va trouver un plan qui est le plus court chemin.



Le monde réel n'est pas parfait !

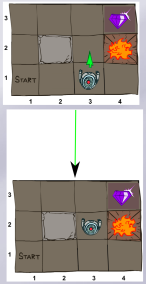
Environnement partiellement observable

Les perceptions sont incertaines.

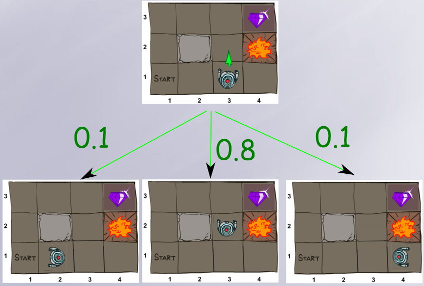
Environnement stochastique (non-déterministe)

Le résultat d'une action est incertain.

environnement déterministe

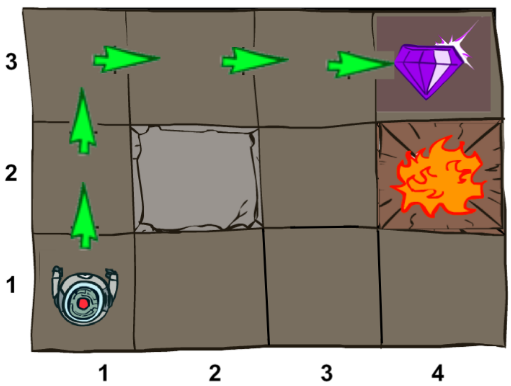


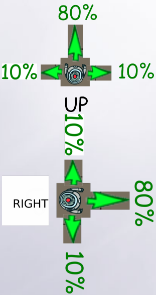
environnement stochastique



Limites de la planification a priori

On modifie les règles ...



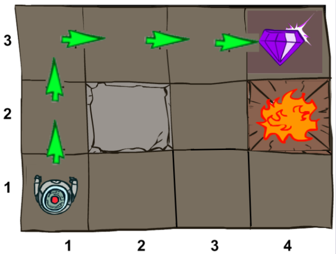


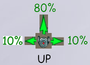
les 4 actions sont non-déterministes

Quelle est la probabilité que le plan trouvé par A* (UP UP RIGHT RIGHT RIGHT) atteigne le but dans un environnement stochastique ?

Limites de la planification a priori

Le plan trouvé par A* (UP UP RIGHT RIGHT RIGHT) atteint le but avec une probabilité de 32,776%.



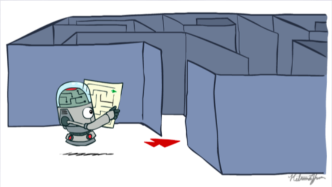


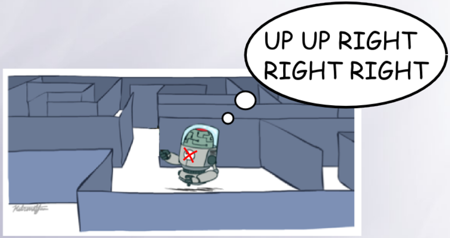
Comment faire mieux ?

Problème de planification a priori

Modèle par planification a priori

1. Planification (A*) calcule un plan = UP UP RIGHT RIGHT RIGHT
2. Exécution du plan « en aveugle » (action sans perception)



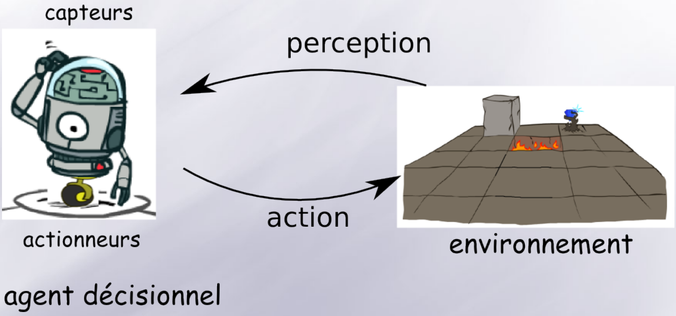


- A* suppose que l'environnement est déterministe.
- A* planifie puis exécute (exécution en aveugle).

Hypothèses d'environnement connu et parfait.

Limites de la planification a priori

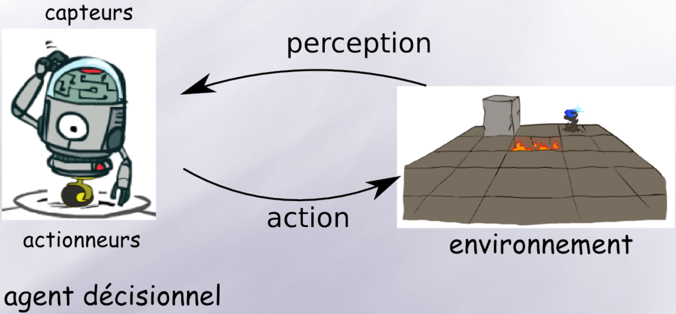
On a besoin d'un feedback !



- entrelacer planification et exécution : *re-planning*
  cf. Darpa challenge Udacity
- utiliser un modèle qui prend en compte les incertitudes et planifie une seule fois

Limites de la planification a priori

On a besoin d'un feedback !



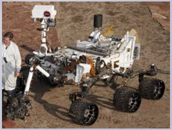
- entrelacer planification et exécution : *re-planning*
  cf. Darpa challenge Udacity
- utiliser un modèle qui prend en compte les incertitudes et planifie une seule fois

Quelques domaines d'applications

- Aérospatiale
- Militaire
- Robotique industrielle, transport
- Vie de tous les jours (aspirateur automatique, tondeuse automatique)
- Informatique (Jeux vidéo, Informatique ambiante)








Plan

Formalisation mathématique

- Environnement stochastique (avec actions incertaines)
- On va tout d'abord définir le **problème** : modèle MDP (*Markov Decisional Process*)
- On va ensuite définir sa **solution** : une politique
- On va ensuite définir l'**objectif** : trouver une politique optimale

Plan

Environnement stochastique sans agent

Chaîne de Markov

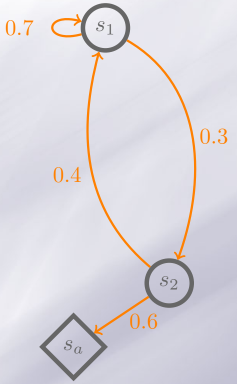
- ensemble fini d'états : S
- fonction de transition $T : S \times S \to [0; 1]$
  $T(s, s') = P(s_{t+1} = s' | s_t = s)$

Propriété de Markov (sans mémoire)

Les transitions ne dépendent que de l'état actuel :
$P(s_{t+1} | s_t, s_{t-1} ..., s_0) = P(s_{t+1} | s_t)$

Ajoutons une composante décisionnelle pour modéliser un agent.

Environnement stochastique sans agent



Chaîne de Markov

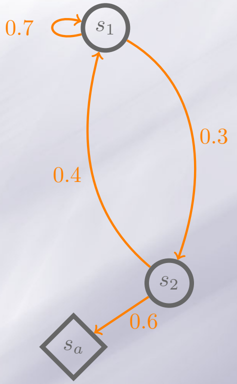
- ensemble fini d'états : S
- fonction de transition $T : S \times S \rightarrow [0;1]$
  $T(s, s') = P(s_{t+1} = s' | s_t = s)$

Propriété de Markov (sans mémoire)

Les transitions ne dépendent que de l'état actuel :
$P(s_{t+1} | s_t, s_{t-1} ..., s_0) = P(s_{t+1} | s_t)$

Ajoutons une composante décisionnelle pour modéliser un agent.

Environnement stochastique sans agent



Chaîne de Markov

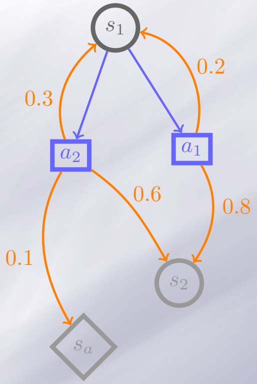
- ensemble fini d'états : S
- fonction de transition $T : S \times S \rightarrow [0; 1]$
  $T(s, s') = P(s_{t+1} = s' | s_t = s)$

Propriété de Markov (sans mémoire)

Les transitions ne dépendent que de l'état actuel :
$P(s_{t+1} | s_t, s_{t-1} ..., s_0) = P(s_{t+1} | s_t)$

Ajoutons une composante décisionnelle pour modéliser un agent.

Environnement stochastique avec agent



Processus Décisionnel Markovien (MDP)

- ensemble fini d'états : S
- ensemble fini d'actions : A ou $A(s)$
- fonction de transition $T : S \times A \times S \rightarrow [0; 1]$
  $T(s, a, s') = P(s_{t+1} = s' | s_t = s, a_t = a)$
- ...

Propriété de Markov

Les conséquences d'une action a_t (fonction de transition) ne dépendent que de l'état courant s_t :

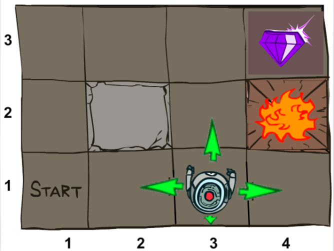
$$P(s_{t+1}|s_t, a_t, r_t, s_{t-1}, a_{t-1}, ..., s_0, a_0) = P(s_{t+1}|s_t, a_t)$$

L'agent n'agit qu'en fonction de son état courant.

Dans notre problème . . .

Processus Décisionnel markovien (MDP)

- ensemble fini d'états : S
- ensemble fini d'actions : A ou $A(s)$
- fonction de transition $T : S \times A \times S \to [0; 1]$
  $T(s, a, s') = P(s_{t+1} = s' | s_t = s, a_t = a)$

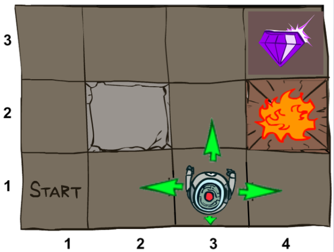


- $|S| =?$
- $A =?$
-

Dans notre problème . . .

Processus Décisionnel markovien (MDP)

- ensemble fini d'états : S
- ensemble fini d'actions : A ou $A(s)$
- fonction de transition $T : S \times A \times S \to [0;1]$
  $T(s,a,s') = P(s_{t+1} = s' | s_t = s, a_t = a)$

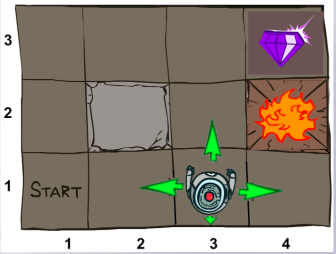


- $|S| = 11$
- $A = UP, DOWN, RIGHT, LEFT$
-

Dans notre problème . . .

Processus Décisionnel markovien (MDP)

- ensemble fini d'états : S
- ensemble fini d'actions : A ou $A(s)$
- fonction de transition $T : S \times A \times S \rightarrow [0;1]$
  $T(s,a,s') = P(s_{t+1} = s' | s_t = s, a_t = a)$

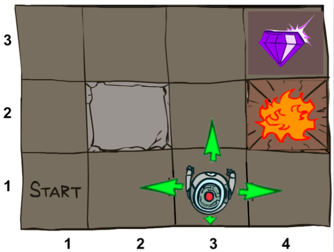


- $|S| = 11$
- $A = UP, DOWN, RIGHT, LEFT$
- $T(1 \times 1, UP, 1 \times 2) = ?$
  $T(1 \times 1, UP, 2 \times 1) = ?$

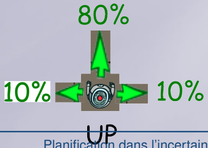
Dans notre problème ...

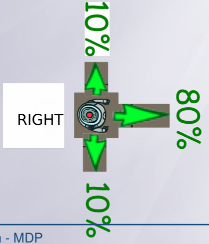
Processus Décisionnel markovien (MDP)

- ensemble fini d'états : S
- ensemble fini d'actions : A ou $A(s)$
- fonction de transition $T : S \times A \times S \to [0;1]$
  $T(s,a,s') = P(s_{t+1} = s' | s_t = s, a_t = a)$



- $T(1\times 1, UP, 1\times 2) = 0.8$
  $T(1\times 1, UP, 2\times 1) = 0.1$
  $T(1\times 1, UP, 1\times 1) = 0.1$
- $T(1\times 1, RIGHT, 2\times 1) = 0.8$
  $T(1\times 1, RIGHT, 1\times 1) = 0.1$ ...

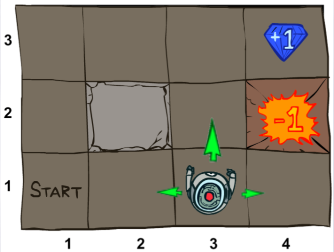




Dans notre problème . . .

Processus Décisionnel Markovien (MDP)

- ensemble fini d'états : S
- ensemble fini d'actions : A ou $A(s)$
- fonction de transition $T : S \times A \times S \rightarrow [0; 1]$
- fonction de renforcement $R : S \times A \times S \rightarrow \mathbb{R}$

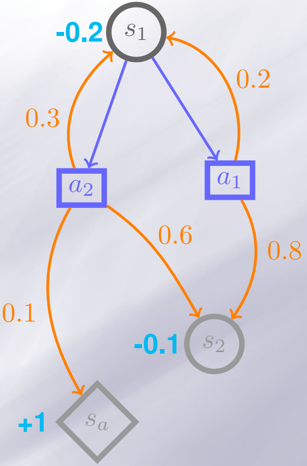


- $|S| = 11$
- $A = UP, DOWN, RIGHT, LEFT$
- $R(3\text{x}3, RIGHT, 4\text{x}3) = 1$
- $R(3\text{x}3, DOWN, 4\text{x}3) = 1$
- $R(3\text{x}2, RIGHT, 4\text{x}2) = -1$

Définition de l'objectif

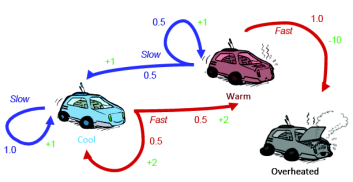
Les récompenses indiquent à quoi aboutir mais pas comment y parvenir.

Conclusion : modèle MDP



Exemple1 d'un MDP

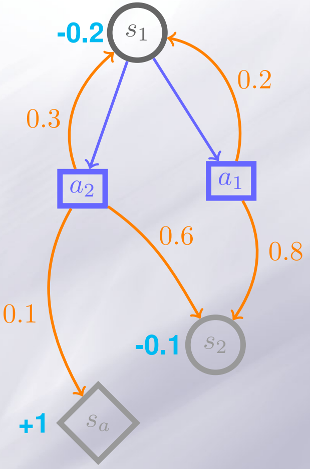
- On a défini le problème (MDP) :
  - ensemble fini d'états : S (s_1, s_2, s_a dans exemple 1)
  - ensemble fini d'actions : A (a_1, a_2 dans exemple 1)
  - fonction de transition $T : S \times A \times S \rightarrow [0; 1]$ (en orange dans exemple 1)
  - fonction de renforcement $R : S \times A \times S \rightarrow \mathbb{R}$ (en bleu dans exemple 1)

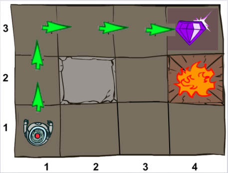


Exemple2 d'un MDP (3 états, 2 actions)

Plan

Solution d'un MDP





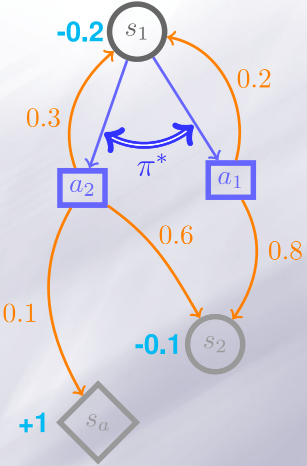
Solution / Environnement :

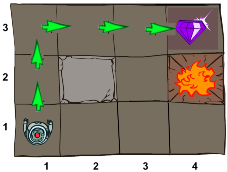
plan / déterministe

politique / stochastique

Politique $\pi : S \rightarrow A$: Fonction qui associe une (ou plusieurs) action(s) à exécuter dans tout état

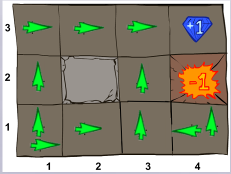
Solution d'un MDP





Solution / Environnement :

plan / déterministe



politique / stochastique

Politique $\pi : S \rightarrow A$: Fonction qui associe une (ou plusieurs) action(s) à exécuter dans tout état

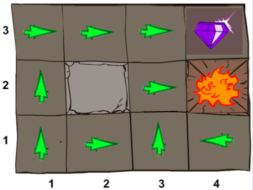
Familles de politiques pour les MDP

politique π_t	déterministe	stochastique
markovienne	$s_t \to a_t$	$s_t, a_t \to [0,1]$
histoire-dépendante	$h_t \to a_t$	$h_t, a_t \to [0,1]$

- $h_t = (s_0, a_0, ..., a_{t-1}, a_{t-1}, s_t)$

Politiques stationnaires

- Si le choix de la meilleure décision à prendre dépend de l'instant t, la politique est non-stationnaire π_t.
- Sinon la politique est stationnaire $\forall t\ \pi_t = \pi$



politique markovienne déterministe stationnaire $\pi : S \to A$

Plan

Solution optimale d'un MDP

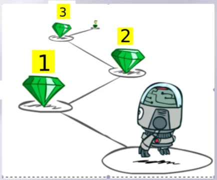
Résoudre un MDP consiste à trouver une politique optimale notée π^*.

Politique optimale π^*

Donne pour tout état, l'action permettant de maximiser les récompenses que l'on espère obtenir à travers la séquence d'états futurs.

Politique optimale déterministe $\pi^* : S \rightarrow A$ maximise l'espérance de :

$$G([r_1, r_2, r_3, ...]) = r_1 + r_2 + r_3 + ... = \sum_{t=0}^{T} r_{t+1}$$



Horizon T

Nombre de pas de temps sur lesquels l'agent raisonne (phase 1 : planification) pour prendre ses décisions. T peut être fini ou infini.

Solution optimale d'un MDP

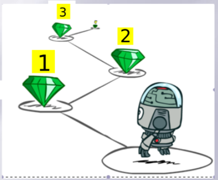
Résoudre un MDP consiste à trouver une politique optimale notée π^*.

Politique optimale π^*

Donne pour tout état, l'action permettant de maximiser les récompenses que l'on espère obtenir à travers la séquence d'états futurs.

Politique optimale déterministe $\pi^* : S \rightarrow A$ maximise l'espérance de :

$$G([r_1, r_2, r_3, ...]) = r_1 + r_2 + r_3 + ... = \sum_{t=0}^{T} r_{t+1}$$



Horizon T

Nombre de pas de temps sur lesquels l'agent raisonne (phase 1 : planification) pour prendre ses décisions. T peut être fini ou infini.

Solution optimale d'un MDP

Résoudre un MDP consiste à trouver une politique optimale notée π^*.

Politique optimale π^*

Donne pour tout état, l'action permettant de maximiser les récompenses que l'on espère obtenir à travers la séquence d'états futurs.

Politique déterministe $\pi : S \rightarrow A$ qui maximise l'espérance de :

$$G([r_1, r_2, r_3, ...]) = r_1 + r_2 + r_3 + ... = \sum_{t=0}^{T} r_{t+1}$$

Attention !

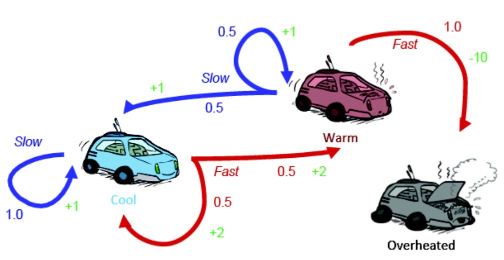
- La politique optimale dépend de comment sont définies les récompenses
- La politique de l'agent dépend de l'interprétation des récompenses
- Il faut bien choisir les récompenses ...

MDP : Un exemple

Processus Décisionnel markovien (MDP)

- ensemble fini d'états : S
- ensemble fini d'actions : A ou $A(s)$
- fonction de transition $T : S \times A \times S \rightarrow [0;1]$
- fonction de renforcement $R : S \times A \times S \rightarrow \mathbb{R}$

3 états, 2 actions (en bleu et rouge)



Quel va être le comportement de l'agent s'il suit la politique optimale ?

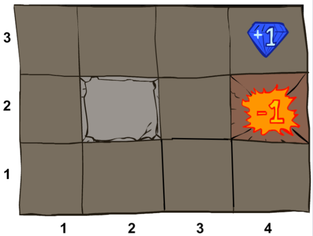
Dans notre problème . . .

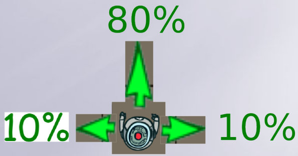
Objectif

Trouver la politique optimale $\pi^* : S \rightarrow A$ qui pour tout état donne l'action permettant de maximiser les récompenses que l'on espère obtenir à travers la séquence d'états futurs

Quelle serait la politique optimale **à horizon infini** (avec $R(*, *, s) = 0 \; \forall s$ non absorbant) ?

Pour chaque $s \in S$, la politique optimale donne l'action qui maximise l'espérance des récompenses futures depuis cet état : $E\{\sum_{t=0}^{\infty} r_{t+1} | \pi, s_0 = s\}$





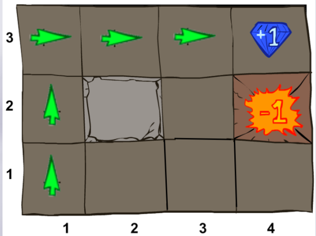
Dans notre problème . . .

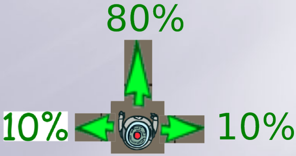
Objectif

Trouver la politique optimale $\pi^* : S \rightarrow A$ qui pour tout état donne l'action permettant de maximiser les récompenses que l'on espère obtenir à travers la séquence d'états futurs

Quelle serait la politique optimale **à horizon infini** (avec $R(*, *, s) = 0 \; \forall s$ non absorbant) ?

Pour chaque $s \in S$, la politique optimale donne l'action qui maximise l'espérance des récompenses futures depuis cet état : $E\{\sum_{t=0}^{\infty} r_{t+1} | \pi, s_0 = s\}$





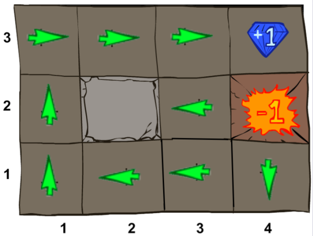
Dans notre problème . . .

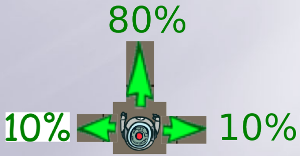
Objectif

Trouver la politique optimale $\pi^* : S \rightarrow A$ qui pour tout état donne l'action permettant de maximiser les récompenses que l'on espère obtenir à travers la séquence d'états futurs

Quelle serait la politique optimale **à horizon infini** (avec $R(*, *, s) = 0 \ \forall s$ non absorbant) ?

Pour chaque $s \in S$, la politique optimale donne l'action qui maximise l'espérance des récompenses futures depuis cet état : $E\{\sum_{t=0}^{\infty} r_{t+1} | \pi, s_0 = s\}$

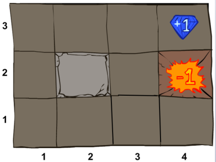


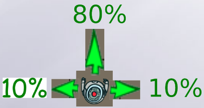


Influence des récompenses

Changer $R \implies \pi^*$ change

Quelle serait la politique optimale à horizon infini avec $R(*, *, s) = -0.1 \; \forall s$ non absorbant ?

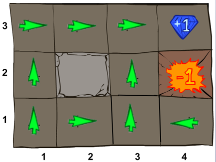


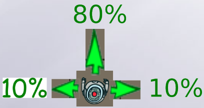


Influence des récompenses

Changer $R \implies \pi^*$ change

Quelle serait la politique optimale à horizon infini avec $R(*, *, s) = -0.1 \ \forall s$ non absorbant ?

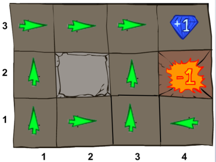


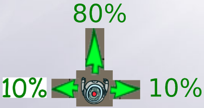


Influence des récompenses

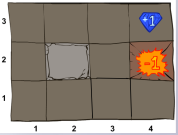
Changer $R \implies \pi^*$ change

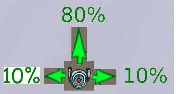
Quelle serait la politique optimale à horizon infini avec $R(*,*,s) = -0.1\ \forall s$ non absorbant ?

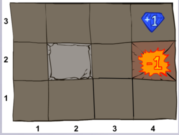


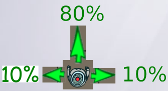


Quelle serait la politique optimale à horizon infini avec $\forall s$ non absorbant $R(*,*,s) = -2$? $+2$?





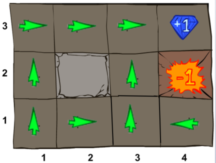


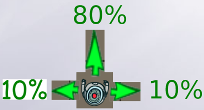


Influence des récompenses

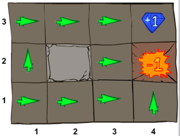
Changer $R \implies \pi^*$ change

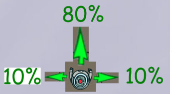
Quelle serait la politique optimale à horizon infini avec $R(*, *, s) = -0.1 \; \forall s$ non absorbant ?

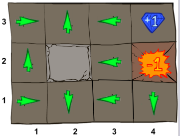


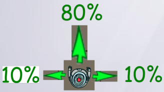


Quelle serait la politique optimale à horizon infini avec $\forall s$ non absorbant $R(*, *, s) = -2$? $+2$?

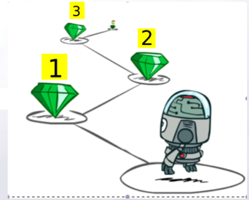








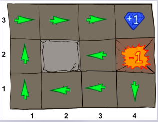
Horizon et politique non stationnaire

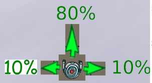


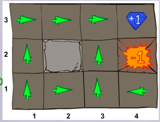
Horizon

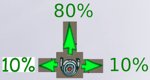
Nombre de pas de temps sur lesquels l'agent raisonne pour prendre ses décisions. Peut être fini ou infini

- Avec horizon infini, la politique est stationnaire $\pi(s) \to a$
- Avec horizon fini, la politique n'est plus stationnaire $\pi(s, t) \to a$









$R(s) = 0.0 \; \forall s$ non absorbant
Horizon fini : π^* évolue au cours du temps

On va s'intéresser au cas avec horizon infini et politique stationnaire.

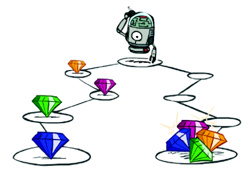
Interprétation des récompenses

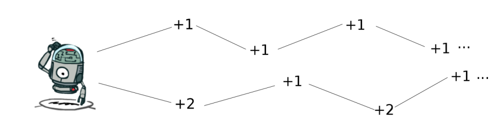
Une politique optimale maximise un critère de performance donné.

Critère total G

Cumul des récompenses instantanées le long d'une trajectoire avec horizon infini :

- $G([r_1, r_2, r_3, ...]) = r_1 + r_2 + r_3 + ... = \sum_{t=0}^{\infty} r_{t+1}$





Temporal credit assignment problem

Quelle séquence de récompenses est préférée selon le critère total ?

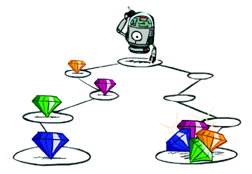
Interprétation des récompenses

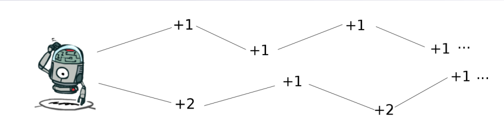
Une politique optimale maximise un critère de performance donné.

Critère total G

Cumul des récompenses instantanées le long d'une trajectoire avec horizon infini :

- $G([r_1, r_2, r_3, ...]) = r_1 + r_2 + r_3 + ... = \sum_{t=0}^{\infty} r_{t+1}$





Quelle séquence de récompenses est préférée selon le critère total ?

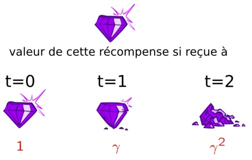
A horizon infini (ou très grand), un compromis est nécessaire entre récompenses immédiates et futures pour borner G.

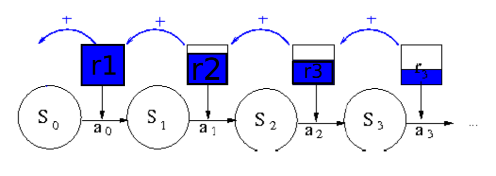
Récompenses pondérées

Critère pondéré G^γ

Cumul des récompenses avec facteur d'atténuation $\gamma \in [0; 1[$

- $G^\gamma([r_1, r_2, r_3, ...]) = r_1 + \gamma r_2 + \gamma^2 r_3 + ... = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$



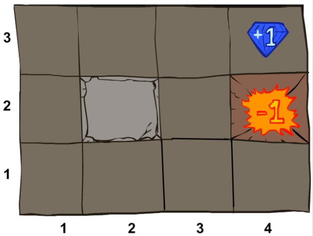


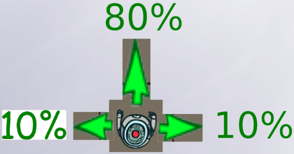
γ règle l'importance des récompenses futures vs. récompenses immédiates.

- $\gamma = 0$: agent cigale qui maximise la récompense immédiate $G([r_1, r_2, r_3, ...]) = r_1$ (horizon de 1)
- $\gamma \rightarrow 1$: agent fourmi qui sacrifie petit gain à court terme pour privilégier meilleur gain à long terme
- $\gamma < 1 \implies G \leq \frac{R_{max}}{1-\gamma}$

Dans notre problème . . .

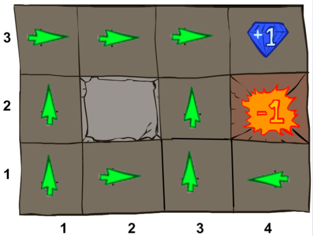
Quelle serait la politique optimale avec un critère pondéré $\gamma = 0.9$ et $R(*, *, s) = 0 \; \forall s$ non absorbant ?

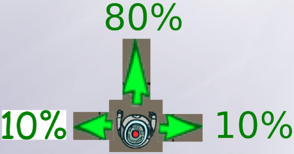




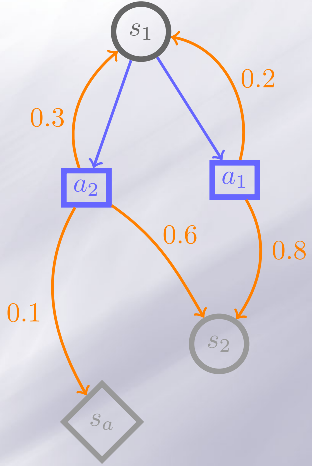
Dans notre problème ...

Quelle serait la politique optimale avec un critère pondéré $\gamma = 0.9$ et $R(*, *, s) = 0 \; \forall s$ non absorbant ?





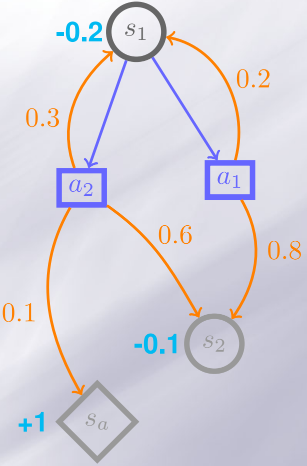
Résumons . . .



Processus Décisionnel Markovien

- Environnement stochastique (avec actions incertaines)
- Récompense r_t à chaque pas de temps (formalise l'objectif)
- Solution = politique optimale π^*
- Maximise critère pondéré avec γ : recompenses futures *vs* immédiates

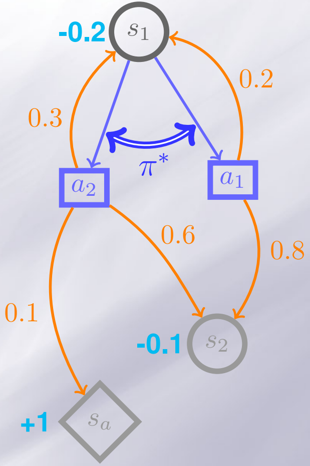
Résumons . . .



Processus Décisionnel Markovien

- Environnement stochastique (avec actions incertaines)
- Récompense r_t à chaque pas de temps (formalise l'objectif)
- Solution = politique optimale π^*
- Maximise critère pondéré avec γ : recompenses futures *vs* immédiates

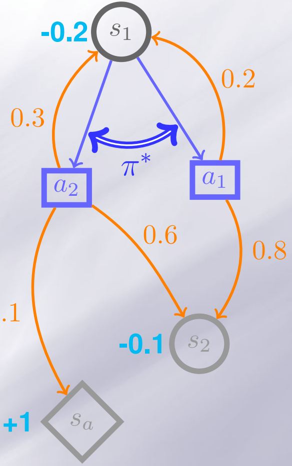
Résumons . . .



Processus Décisionnel Markovien

- Environnement stochastique (avec actions incertaines)
- Récompense r_t à chaque pas de temps (formalise l'objectif)
- Solution = politique optimale π^*
- Maximise critère pondéré avec γ : recompenses futures *vs* immédiates

Résumons . . .

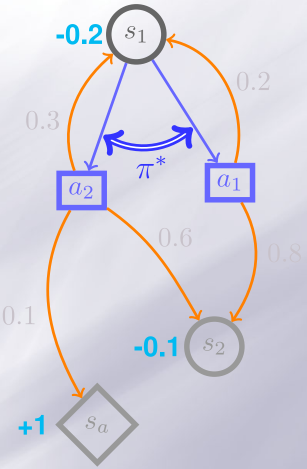


Processus Décisionnel Markovien

- Environnement stochastique (avec actions incertaines)
- Récompense r_t à chaque pas de temps (formalise l'objectif)
- Solution = politique optimale π^*
- Maximise critère pondéré avec γ : recompenses futures *vs* immédiates

Les politiques obtenues sont plus robustes aux incertitudes que les plans obtenus par des méthodes déterministes

Résumons ...



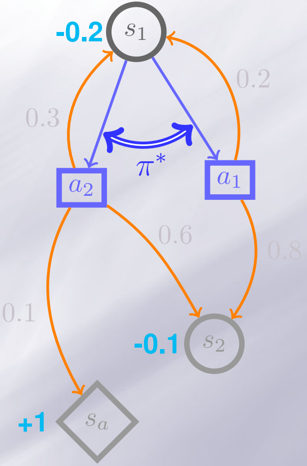
Processus Décisionnel Markovien

- Environnement stochastique (avec actions incertaines)
- Récompense r_t à chaque pas de temps (formalise l'objectif)
- Solution = politique optimale π^*
- Maximise critère pondéré avec γ : recompenses futures *vs* immédiates

Les politiques obtenues sont plus robustes aux incertitudes que les plans obtenus par des méthodes déterministes

Exercice du TD : modélisation sous forme d'un MDP

Résumons . . .



Processus Décisionnel Markovien

- Environnement stochastique (avec actions incertaines)
- Récompense r_t à chaque pas de temps (formalise l'objectif)
- Solution = politique optimale π^*
- Maximise critère pondéré avec γ : recompenses futures *vs* immédiates

Les politiques obtenues sont plus robustes aux incertitudes que les plans obtenus par des méthodes déterministes

Comment calculer la politique optimale ?

Plan

Fonction de valeur

Critère à maximiser

- $G^\gamma([r_1, r_2, r_3, ...]) = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$
- Trouver la politique optimale π^* : $S \rightarrow A$ qui le maximise :

$$\pi^* = \arg\max_{\pi} E\{G^\gamma | \pi\}$$

Fonction de valeur V

- $R(s)$ évalue l'**intérêt immédiat** d'être dans un état s
- $V^\pi(s)$ évalue l'**intérêt sur le long terme** de suivre une politique π à partir de l'état s *i.e.* le retour espérée si on suit π depuis s :

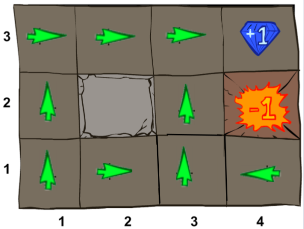
$$V^\pi(s) = E\{\sum_{t=0}^{\infty} \gamma^t r_{t+1} | \pi, s_0 = s\}$$

Illustration de la fonction de valeur

Fonction de valeur V

- $V^\pi(s)$ évalue l'intérêt sur le long terme de suivre une politique π à partir de l'état s *i.e.* **le retour espérée** si on suit π depuis s :

$$V^\pi(s) = E\{\sum_{t=0}^{\infty} \gamma^t r_{t+1} | \pi, s_0 = s\}$$



π

V^π

(exemple avec un environnement déterministe)

Calcul de V^π à partir de π ?

Propriété fondamentale de V

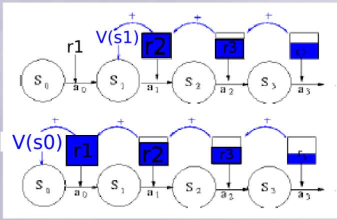
Equation de Bellman

Elle définit la valeur d'un état en fonction de la valeur des états lui succédant :

$$V^\pi(s) = E\{\sum_{t=0}^{\infty} \gamma^t r_{t+1} | \pi, s_0 = s\} = E\{r_1 + \gamma \sum_{t=0}^{\infty} \gamma^t r_{t+2} | \pi, s_0 = s\}$$

$$= \sum_{s' \in S} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma E\{\sum_{t=0}^{\infty} \gamma^t r_{t+2} | \pi, s_1 = s'\}]$$

$$V^\pi(s) = \sum_{s' \in S} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V^\pi(s')]$$



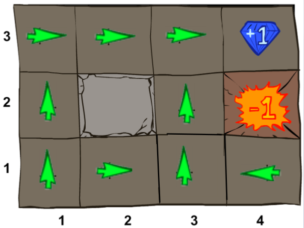
$$V(s_0) = r_1 + \gamma V(s_1)$$

Propriété fondamentale de V

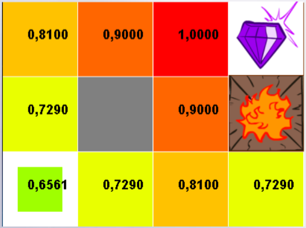
Equation de Bellman

Elle définit la valeur d'un état en fonction de la valeur des états lui succédant :

$$V^\pi(s) = \sum_{s' \in S} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V^\pi(s')]$$



π



V^π

$$\begin{aligned} V(3 \times 2) = {} & 0.8[R(3 \times 2, UP, 3 \times 3) + \gamma V(3 \times 3)] \\ & + 0.1[R(3 \times 2, UP, 4 \times 2) + \gamma V(4 \times 2)] \\ & + 0.1[R(3 \times 2, UP, 3 \times 2) + \gamma V(3 \times 2)] \end{aligned} \quad (1)$$

Nouvel objectif

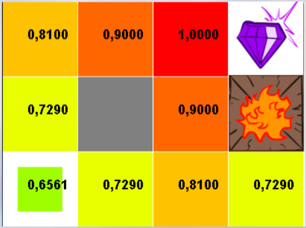
- On peut calculer V^{π} à partir de π
- Mais notre objectif est de trouver π^*

Nouvel objectif

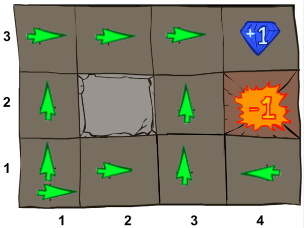
- $V^{\pi^*}(s) = V^*(s) = \max_{\pi} V^{\pi}(s)$
- Si on connait V^*, on peut calculer π^*

$$\pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$$

- Dans chaque état choisit l'action qui va maximiser le retour espéré



V^{π}



π

(exemple avec un environnement déterministe)

Nouvel objectif

- Pour trouver π^* on va chercher V^*
- V^* est l'unique solution de (Equation d'optimalité de Bellman) :

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$$

On veut V^* pour en extraire π^*

Comment calculer V^* ?

- Résoudre le système de $n = |S|$ équations non-linéaires, n inconnues
- Approximation par itération

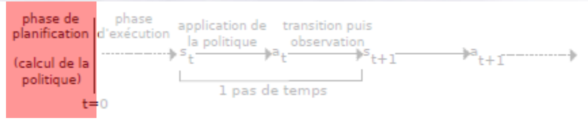
Principe d'optimalité de Bellman

- Si l'on sait trouver la solution optimale (politique) à partir de l'étape $t+1$ quel que soit l'état s_{t+1}, alors on peut trouver la décision optimale à l'étape t pour tout état possible s_t (et donc travailler par récurrence).

Plan

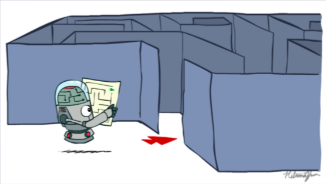
Planifier dans les MDP

Trouver une politique pour accomplir un objectif donné au sein d'un environnement particulier.



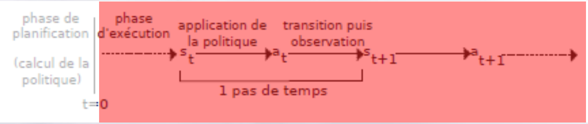
2 phases :

- phase 1 : hors-ligne, planification : calcul de la politique π, l'agent n'agit pas dans l'environnement



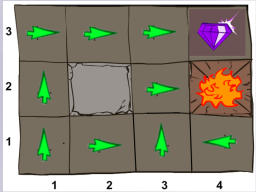
Planifier dans les MDP

Trouver une politique pour accomplir un objectif donné au sein d'un environnement particulier.



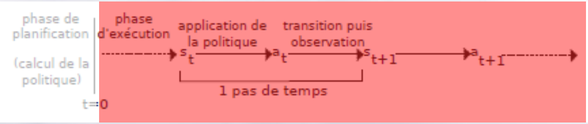
2 phases :

- phase 2 : en-ligne, exécution de la politique : l'agent agit dans l'environnement en suivant la politique calculée



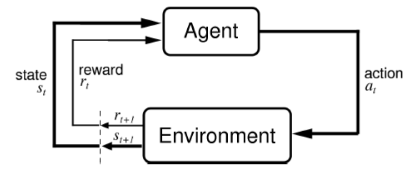
Planifier dans les MDP

Trouver une politique pour accomplir un objectif donné au sein d'un environnement particulier.



2 phases :

- phase 2 : en-ligne, exécution de la politique :
  - Boucle de vie de l'agent : perçoit (s_t,r_t), décide ($\pi(s_t)$) et agit (a_t)
  - processus séquentiel $s_0, a_0, s_1, r_1, a_1, s_2, r_2, a_2, \ldots$



Approximation par itération

Algorithmes pour le calcul de la politique optimale π^* pendant la phase hors-ligne de planification.

Programmation dynamique

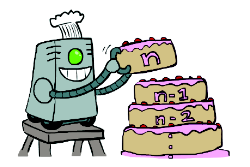
Algorithmes qui calculent V^* ou π^* à partir du modèle de l'environnement (MDP connu) :

- *value iteration* : résolution directe de l'équation d'optimalité de Bellman par évaluation itérative de V^*
- *policy iteration* : construction itérative de π^*

Itérations sur les valeurs

Value iteration [Bellman,1957]

Évaluation itérative de V^* : calcule $V_0 \rightarrow V_1 \rightarrow V_2 ...$



- Initialisation arbitraire de $V_0(s) \forall s$
- Mise à jour de $V_k(s)$ $\forall s$ en utilisant les valeurs estimées à $k-1$ des états voisins (*bootstrap*)

$$V_k(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_{k-1}(s')]$$

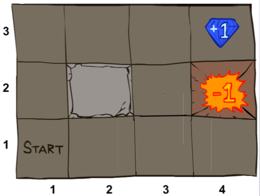
- Répète jusqu'à convergence
  critère d'arrêt $\max_{s \in S} |V_k(s) - V_{k+1}(s)| < \epsilon$

- théorème du point fixe de Banach : convergence de l'algorithme vers V^* solution de l'équation de Bellman pour tout V_0
- complexité de chaque itération $O(S^2 A)$

Itérations sur les valeurs : Illustration

$\forall s$ non absorbant

$V_k(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_{k-1}(s')] \quad \gamma = 0.9$



0,0000	0,0000	0,0000	0,0000
0,0000		0,0000	0,0000
0,0000	0,0000	0,0000	0,0000

V_0

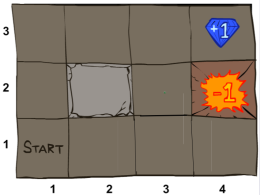
Exercice dans un labyrinthe déterministe

- Itération 0, $k = 0$, $V_k(s) = V_0(s) = 0 \ \forall s$
- Itération 1, $k = 1$, $V_1(s)$?
- Itération 2, $k = 2$, $V_2(s)$?
- Itération 3, $k = 3$, $V_3(s)$?

Itérations sur les valeurs : Illustration

$\forall s$non absorbant $\quad \gamma = 0.9$

$V_k(s) \leftarrow \max\limits_{a} \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_{k-1}(s')]$



100%

0,0000	0,0000	0,0000	0,0000
0,0000		0,0000	0,0000
0,0000	0,0000	0,0000	0,0000

V_0

Pour $s =< 3,3 >$, $k = 1$:

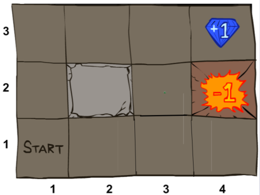
- $a = RIGHT$, $s' =< 4,3 > : 1 \times [1 + \gamma \times 0] = 1$
- $a = LEFT$, $s' =< 2,3 > : 1 \times [0 + \gamma \times 0] = 0$
- $a = UP$, $s' =< 3,3 > : 1 \times [0 + \gamma \times 0] = 0$
- $a = DOWN$, $s' =< 3,2 > : 1 \times [0 + \gamma \times 0] = 0$

$V_1(< 3,3 >) = max\{1,0,0,0\} = 1$

Itérations sur les valeurs : Illustration

$\forall s$ non absorbant $\quad \gamma = 0.9$

$$V_k(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_{k-1}(s')]$$



0,0000	0,0000	0,0000	0,0000
0,0000		0,0000	0,0000
0,0000	0,0000	0,0000	0,0000

V_0

Pour $s =< 3,2 >$, $k = 1$:

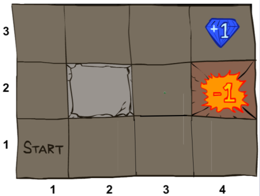
- $a = RIGHT,\ s' =< 4,2 > : 1 \times [-1 + \gamma \times 0] = -1$
- $a = LEFT,\ s' =< 3,2 > : 1 \times [0 + \gamma \times 0] = 0$
- $a = UP,\ s' =< 3,3 > : 1 \times [0 + \gamma \times 0] = 0$
- $a = DOWN,\ s' =< 3,1 > : 1 \times [0 + \gamma \times 0] = 0$

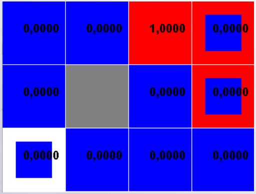
$V_1(< 3,2 >) = max\{-1, 0, 0, 0\} = 0$

Itérations sur les valeurs : Illustration

$\forall s$ non absorbant $\quad \gamma = 0.9$

$V_k(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_{k-1}(s')]$





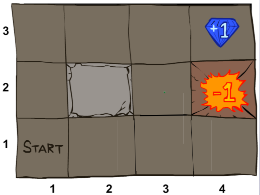
V_1

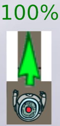
- $V_1(<3,3>) = max\{1,0,0,0\} = 1$
- $V_1(<3,2>) = max\{-1,0,0,0\} = 0$
- Pour les autres s, $\forall a \in A$, $R(s,a,s')$ et $V_0(s') = 0$ donc $V_1(s) = 0$.

Itérations sur les valeurs : Illustration

$\forall s$non absorbant $\quad \gamma = 0.9$

$$V_k(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_{k-1}(s')]$$





0,0000	0,0000	1,0000	0,0000
0,0000		0,0000	0,0000
0,0000	0,0000	0,0000	0,0000

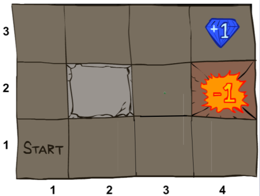
V_1

A vous de calculer V_2 !

Itérations sur les valeurs : Illustration

$\forall s$non absorbant $\quad \gamma = 0.9$

$$V_k(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_{k-1}(s')]$$



100%

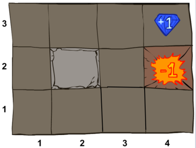
0,8100	0,9000	1,0000	0,0000
0,7290		0,9000	0,0000
0,6561	0,7290	0,8100	0,7290

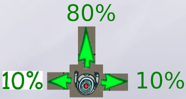
V_5

Itérations sur les valeurs : Illustration

$\forall s$ non absorbant $\quad \gamma = 0.9$

$$V_k(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_{k-1}(s')]$$





0,0000	0,0000	0,0000	0,0000
0,0000		0,0000	0,0000
0,0000	0,0000	0,0000	0,0000

V_0

Exercice dans un labyrinthe stochastique

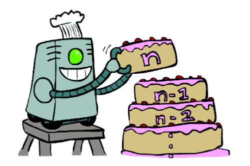
$V_0(s) = 0 \ \forall s$

Calculer $V_1(s)$ et $V_2(s)$ (attention, il y a maintenant plusieurs états d'arrivée (s') pour un s et a !)

Itérations sur les valeurs

Value iteration [Bellman,1957]

Évaluation itérative de V^* : calcule $V_0 \rightarrow V_1 \rightarrow V_2 \ldots$



- Initialisation arbitraire de $V_0(s) \forall s$
- Mise à jour de $V_k(s)$ $\forall s$ en utilisant les valeurs estimées à $k-1$ des états voisins (*bootstrap*)

$$V_k(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s')+\gamma V_{k-1}(s')]$$

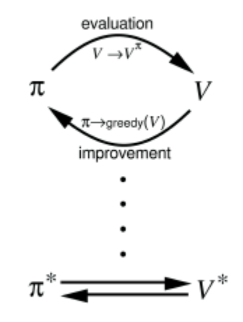
- Répète jusqu'à convergence

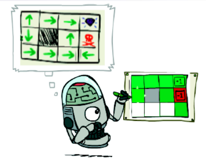
critère d'arrêt $\max_{s \in S} |V_k(s) - V_{k+1}(s)| < \epsilon$

- extraction de π^* gloutonne sur V^*

$$\forall s \in S \quad \pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$$

Itérations sur les politiques





Evaluation d'une politique

Amélioration de politique

Policy iteration [Howard,1960]

- Partant de π_0 quelconque, l'itération de la politique mène à π^* avec succession d'évaluations et d'améliorations :

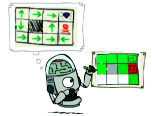
$$\pi_0 \to^e V^{\pi_0} \to^a \pi_1 \to^e V^{\pi_1} \to^a \pi_2 \to^e V^{\pi_2} \ldots \to^a \pi_* \to^e V^*$$

Itérations sur les politiques

Policy iteration [Howard,1960]

- Partant de π_0 quelconque, l'itération de la politique mène à π^* avec succession d'évaluations et d'améliorations :

$$\pi_0 \rightarrow^e V^{\pi_0} \rightarrow^a \pi_1 \rightarrow^e V^{\pi_1} \rightarrow^a \pi_2 \rightarrow^e V^{\pi_2} \ldots \rightarrow^a \pi_* \rightarrow^e V^*$$



- Evaluation d'une politique π : calcul itératif de V^π

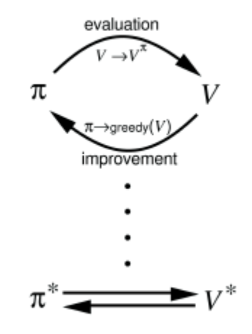
$$V^\pi_{k+1}(s) = \sum_{s' \in S} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V^\pi_k(s')]$$

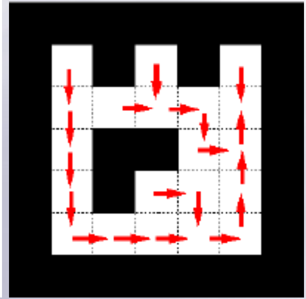
- Amélioration de politique

$$\forall s \in S \quad \pi'(s) \leftarrow \arg\max_a [\sum_{s' \in S} T(s, a, s')[R(s, a, s') + \gamma V^\pi(s')]]$$

Itérations sur les politiques : Exemple

- $|S| = 20$, $A = \{N, S, E, O\}$, transitions déterministes, $R(s_{14}, S) = R(s_{18}, E) = 0.9$, $\gamma = 0.9$

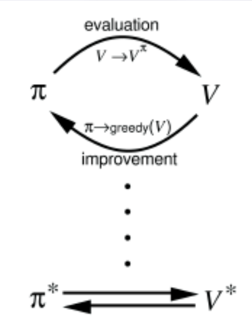


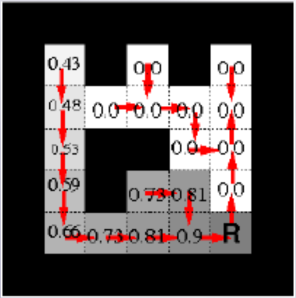


π_0

Itérations sur les politiques : Exemple

- $|S| = 20$, $A = \{N, S, E, O\}$, transitions déterministes, $R(s_{14}, S) = R(s_{18}, E) = 0.9$, $\gamma = 0.9$

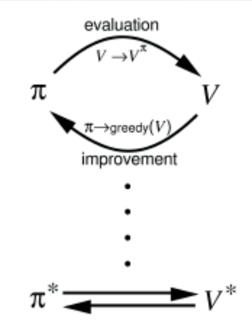


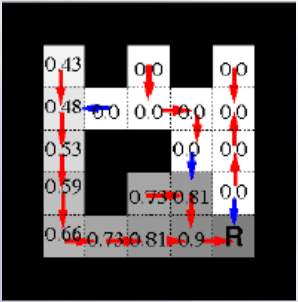


$$\forall s \in S \quad V_0(s) \leftarrow evaluate(\pi_0(s))$$

Itérations sur les politiques : Exemple

- $|S| = 20$, $A = \{N, S, E, O\}$, transitions déterministes, $R(s_{14}, S) = R(s_{18}, E) = 0.9$, $\gamma = 0.9$

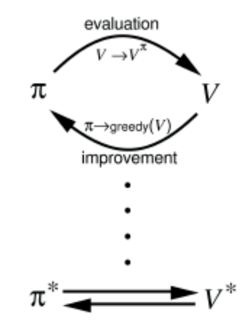


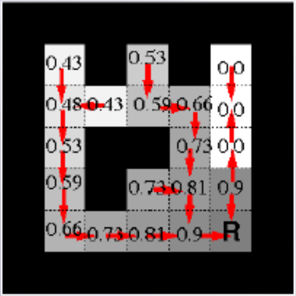


$$\forall s \in S \quad \pi_1(s) \leftarrow ameliore(\pi_0(s), V_0(s))$$

Itérations sur les politiques : Exemple

- $|S| = 20$, $A = \{N, S, E, O\}$, transitions déterministes, $R(s_{14}, S) = R(s_{18}, E) = 0.9$, $\gamma = 0.9$

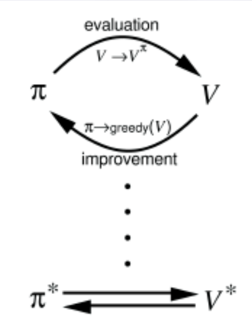


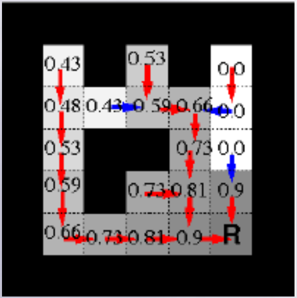


$$\forall s \in S \quad V_1(s) \leftarrow evaluate(\pi_1(s))$$

Itérations sur les politiques : Exemple

- $|S| = 20$, $A = \{N, S, E, O\}$, transitions déterministes, $R(s_{14}, S) = R(s_{18}, E) = 0.9$, $\gamma = 0.9$

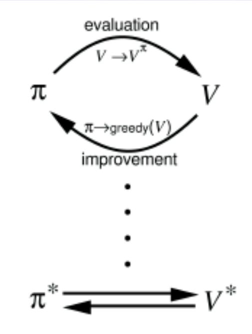


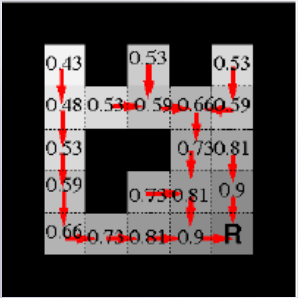


$$\forall s \in S \quad \pi_2(s) \leftarrow ameliore(\pi_1(s), V_1(s))$$

Itérations sur les politiques : Exemple

- $|S| = 20$, $A = \{N, S, E, O\}$, transitions déterministes, $R(s_{14}, S) = R(s_{18}, E) = 0.9$, $\gamma = 0.9$

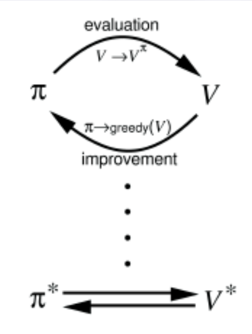


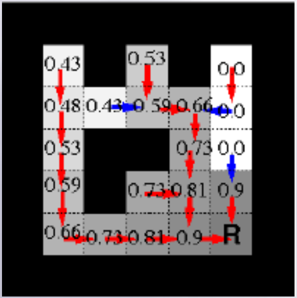


$$\forall s \in S \quad V_2(s) \leftarrow evaluate(\pi_2(s))$$

Itérations sur les politiques : Exemple

- $|S| = 20$, $A = \{N, S, E, O\}$, transitions déterministes, $R(s_{14}, S) = R(s_{18}, E) = 0.9$, $\gamma = 0.9$

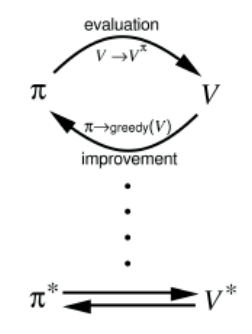


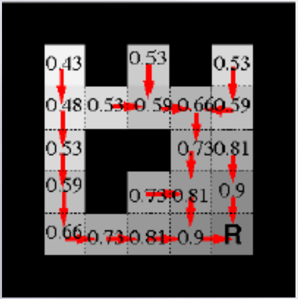


$$\forall s \in S \quad \pi_3(s) \leftarrow ameliore(\pi_2(s), V_2(s))$$

Itérations sur les politiques : Exemple

- $|S| = 20$, $A = \{N, S, E, O\}$, transitions déterministes, $R(s_{14}, S) = R(s_{18}, E) = 0.9$, $\gamma = 0.9$





$$\forall s \in S \quad V_3(s) \leftarrow evaluate(\pi_3(s))$$

Résumons ...

- *policy iteration* et *value iteration* calculent itérativement V^* à partir du modèle MDP connu
- *value iteration* met à jour V jusqu'à convergence puis extrait π
- *policy iteration* alterne évaluation d'une politique fixée (itérations sur les valeurs) et amélioration de politique

Plan

Extensions des MDP

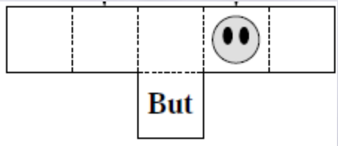
Les méthodes vues jusqu'ici supposent :

- une connaissance parfaite de l'état de l'agent
- une connaissance complète du modèle MDP
- un seul agent

Extensions des MDP

MDP Partiellement Observable

- connaissance imparfaite de l'état de l'agent : incertitudes sur l'état de l'agent



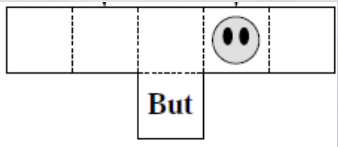
L'agent n'agit qu'en fonction de son observation immédiate $\pi : S \rightarrow A$:

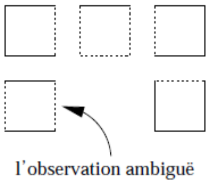
- si l'agent perçoit l'état du labyrinthe dans lequel il se trouve : problème markovien

Extensions des MDP

MDP Partiellement Observable

- connaissance imparfaite de l'état de l'agent : incertitudes sur l'état de l'agent





L'agent n'agit qu'en fonction de son observation immédiate $\pi : S \rightarrow A$:

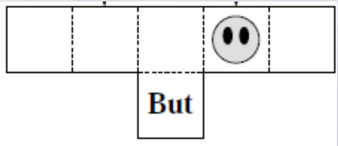
- si l'agent ne perçoit que la présence de murs dans les 4 directions :

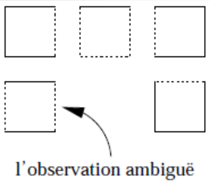
L'agent peut-il agir de manière optimale en fonction de son observation courante ?

Extensions des MDP

MDP Partiellement Observable

- connaissance imparfaite de l'état de l'agent : incertitudes sur l'état de l'agent





L'agent n'agit qu'en fonction de son observation immédiate $\pi : S \rightarrow A$:

- si l'agent ne perçoit que la présence de murs dans les 4 directions :

L'agent peut-il agir de manière optimale en fonction de son observation courante ?

Problème non-markovien, impossible dans les états ambigüs de trouver l'action optimale

Extensions des MDP

MDP Partiellement Observable

- connaissance imparfaite de l'état de l'agent : incertitudes sur l'état de l'agent

POMDP Partially Observable MDP

- $< S, A, T, R >$ MDP "'sous-jacent"
- Ω ensemble d'observations
- $O : S \times \Omega \rightarrow [0; 1]$ fonction d'observation qui donne $P(o_{t+1} = o' | s_{t+1} = s')$
- Solutions : trouver l'historique suffisant pour obtenir des observations étendues markoviennes, recherche directe de la politique dans l'espace des politiques, ...

Extensions des MDP

Connaissance imparfaite du modèle

- T et R inconnus $\rightarrow$ voir cours suivant sur l'apprentissage par renforcement

Extensions des MDP

Systèmes Multi-Agent (SMA)

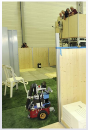
Dec-(PO)MDP [bernstein2002] $< n, S, A, T, R, \Omega, O >$:

- n nombre de robots,
- $s \in S$ état joint du SMA $s = (s_1, s_2, ..., s_n)$
- $a \in A$ action jointe
- $T : S \times A \times S \rightarrow [0; 1]$ fonction de transition des n robots d'un état joint s à s' après exécution de l'action jointe a
- $R : S \rightarrow \Re$ fonction de récompense sur l'état joint s

Plan

Contexte : défi robotique CAROTTE

5 équipes



PACOM

YOJI

CARTOMATIC

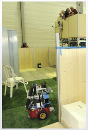
ROBOTS_MALINS

COREBOTS

- Objectif : système robotisé autonome pour la cartographie et l'exploration d'un environnement dynamique et inconnu
- Problématiques :
  - mobilité
  - localisation et cartographie (SLAM)
  - détection d'objets
  - décision : où aller pour explorer efficacement ?

Contexte : défi robotique CAROTTE

5 équipes



PACOM

YOJI

CARTOMATIC

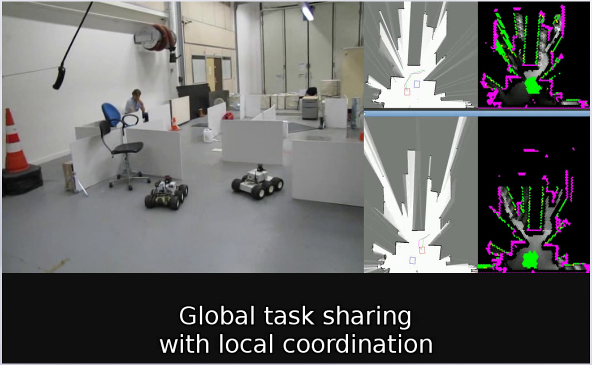
ROBOTS_MALINS

COREBOTS

- Objectif : système robotisé autonome pour la cartographie et l'exploration d'un environnement dynamique et inconnu
- Problématiques :
  - mobilité
  - localisation et cartographie (SLAM)
  - détection d'objets
  - décision : où aller pour explorer efficacement ?

Hypothèses

- Systèmes multi-robots : robots indépendants, pas de station centrale
- SLAM distribué/communication : chaque robot a accès à la carte fusionnée et aux localisations des robots
- reconnaissance “au fil de l'eau”



Objectifs

Développer des stratégies d’exploration multi-robot

Coordination décentralisée d’agents décisionnels :

- coordination globale : allocation des buts d’exploration (couverture efficace de la zone, minimiser les recouvrements).
- interactions locales : à minimiser car exploration non efficace (recouvrements) et conflits nécessitant une coordination locale.

- Modèles de décision multi-agent : MDP multi-agent (MMDP, Dec-MDP, ...)
- Résolution optimale très difficile (même pour 2 agents)
- Proposition d’une méthode de résolution approchée

Extensions des MDP

Systèmes Multi-Agent (SMA)

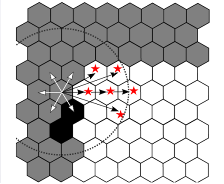
Dec-(PO)MDP [bernstein2002] $< n, S, A, T, R, \Omega, O >$:

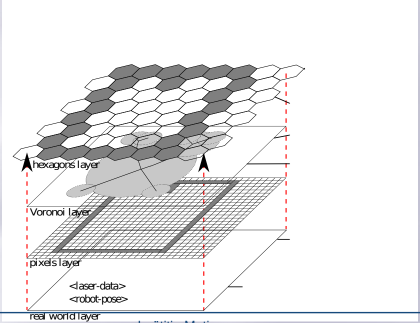
- n nombre de robots,
- $s \in S$ état joint du SMA $s = (s_1, s_2, ..., s_n)$
- $a \in A$ action jointe
- $T : S \times A \times S \to [0; 1]$ fonction de transition des n robots d'un état joint s à s' après exécution de l'action jointe a
- $R : S \to \Re$ fonction de récompense sur l'état joint s

MDP augmenté

Notre approche

1. ensemble de MDPs locaux $\{MDP_1, ..., MDP_n\}$, un par agent.





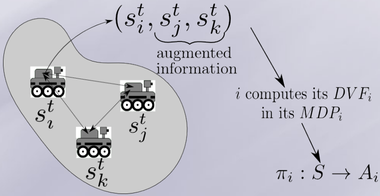
MDP_i :

- S : 6 orientations $\times$ nombre d'hexagones
- A : avance, recule, tourne droite, tourne gauche + suivre voronoi
- T : état espéré suite à une action atteint avec une forte probabilité
- R : propagation des récompenses jusqu'à 2 mètres des frontières

Modèle MDP augmenté

Notre approche

1. ensemble de MDPs locaux $\{MDP_1, ..., MDP_n\}$, un par agent.
2. chaque MDP local est un MDP augmenté par des informations
3. l'information augmentée permet de prédire les intentions des autres par empathie

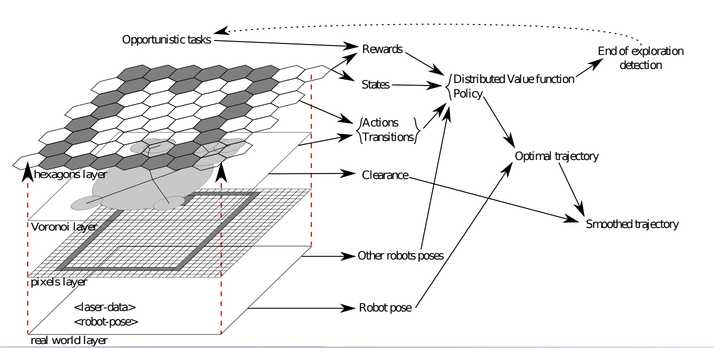


permanent communication

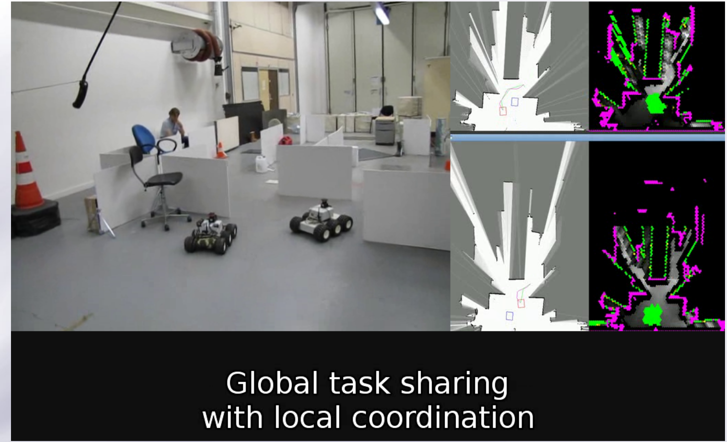
Modèle MDP augmenté

Notre approche

1. chaque robot résout par planification (*value iteration*) son MDP augmenté en utilisant l'information augmentée afin de minimiser les interactions
2. calcul d'un nouveau modèle et nouvelle politique toutes les secondes



Vidéos



Bibliographie

- [Sutton & Barto, 1998] : la bible du domaine
  `https://webdocs.cs.ualberta.ca/~sutton/book/the-book.html`
  version 2017 :
  `http://incompleteideas.net/book/bookdraft2017nov5.pdf`
- [Buffet & Sigaud, 2008] : en français
- [Sigaud & Buffet, 2010] : traduction (améliorée) de 2