GERLAND Loïc

LETOURNEUR Léo

Laëtitia MATIGNON

IA - TP1

Planification Stochastique & MDP

# Polytech

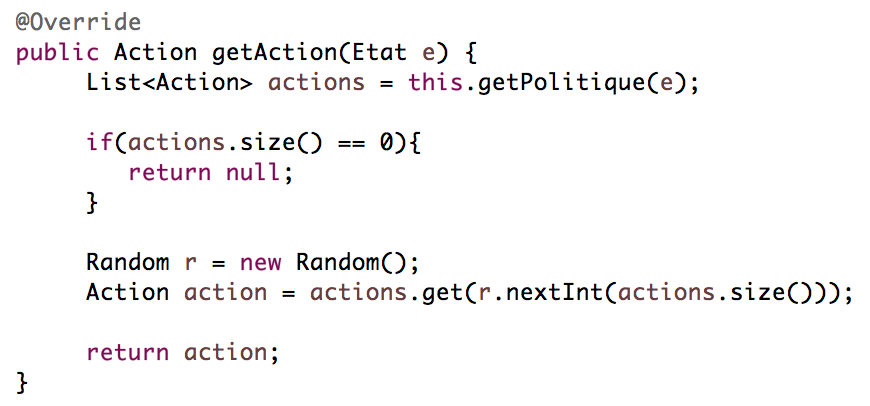
2016

1. Agent Aléatoire

## Question 1

Lorsque le bruit est à 0, l’agent suit exactement les directions souhaitées (déterministe : pour une action, on a toujours le même effet). Lorsque le bruit est supérieur à 0, il arrive que, aléatoirement, l’agent n’effectue pas le déplacement souhaité (environnement stochastique).

## Question 2



Pour savoir quelle action (haut, bas, gauche, droite) doit être effectuée, nous récupérons d’abord les actions possibles avec getPolitique(e). Si aucune action est possible, nous retournons null. Sinon, nous choisissons une action au hasard dans la liste.

2. Algorithme *value iteration*

## Question 1

*Code dans le package.*

**updateV()** permet de mettre à jour la matrice des valeurs pour chaque état en fonction de la matrice à l’itération précédente. La fonction permet aussi de maximiser l’erreur pour gérer la convergence avec la variable *delta*. Il est aussi possible de gérer la valeur maximum vmax et la valeur minimum vmin pour gérer le gradient (couleur) des états.

**getAction(Etat e)** permet de choisir une action dans la politique de l’état actuel. Si plusieurs actions ont la même valeur, la direction choisie est aléatoire parmi ces actions.

**getPolitique(Etat \_e)** permet de calculer la valeur des actions possibles dans l’état actuel. On retourne une liste : Les actions qui ont la plus forte valeur.

## Question 2

Ci-dessous, quelques itérations sur le *BookGrid* vu en cours, avec l’algorithme *value iteration*:

|  |  |  |
| --- | --- | --- |
| **Itération** | **Valeurs** | **Politique** |
| 1 | IA/Matignon/Itérations/Nombre-1.png | IA/Matignon/Itérations/Politique-1.png |
| 2 | IA/Matignon/Itérations/Nombre-2.png | IA/Matignon/Itérations/Politique-2.png |
| 3 | IA/Matignon/Itérations/Nombre-3.png | IA/Matignon/Itérations/Politique-3.png |
| Convergence | IA/Matignon/Itérations/Nombre-Convergence.png | IA/Matignon/Itérations/Politique-Convergence.png |

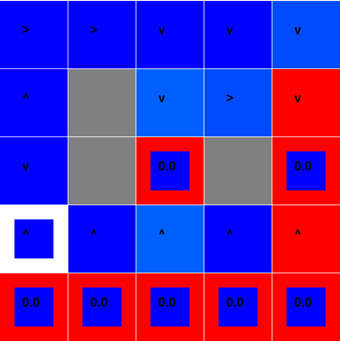
Nous pouvons voir d’après les itérations précédentes, que nos résultats sont identiques à ceux du cours.

3. Influence des paramètres

## Question 1

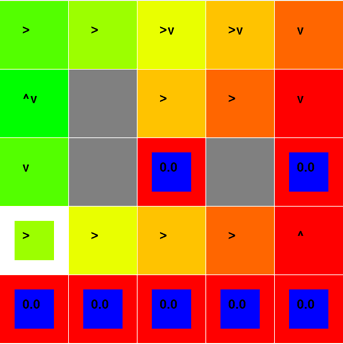
Pour un bruit inférieur ou égal à 0.01, l’agent va traverser le pont. La récompense +10 n’est pas assez élevée pour que l’agent traverse le pont et risque des récompenses de -100. Il faut donc que le bruit soit assez petit pour que la direction prise par l’agent soit la bonne.

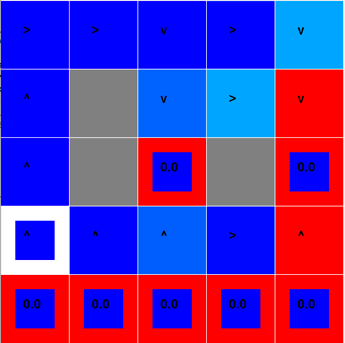
## Question 2

1. Chemin risqué pour récompense +1 :

En utilisant un gamma à 0.1, l’agent ne voit pas le chemin par le haut et n’aura aucune chance de monter. La seule solution possible, est de converger vers le +1 par le bas. On peut voir ci à droite que la politique permet de ne pas partir vers le haut et de converger sur l’état absorbant +1.

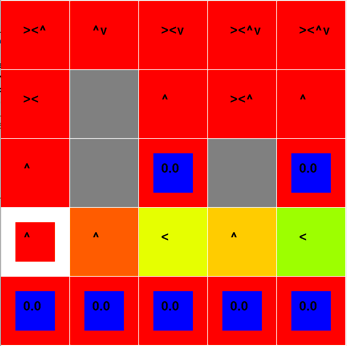
2. Chemin risqué pour récompense +10 :

Avec un bruit nul, l’agent va choisir le plus court chemin (qui est aussi le plus risqué) car il n’a aucune chance de se tromper. On peut voir ci à droite que la politique permet de partir sur la droite jusqu’à l’état absorbant +10.

3. Chemin sûr pour récompense +1 :

Avec un gamma égal à 0.2, l’agent ne voit pas assez loin pour aller chercher le +10, et converge donc vers le +1 en empruntant le chemin le plus sûr. On peut voir ci à droite que la politique permet de monter dans la partie sûre et de redescendre par le haut sur l’état absorbant +1.

4. Eviter les états absorbants :

Il suffit d’ajouter une récompense supérieure aux maximum des états absorbant (ici 10), pour que l’agent préfère rester sur les états non absorbants. On peut voir ci à droite que la politique permet de ne jamais aller sur un état absorbant.

4. Modélisation d’un problème en MDP

## Question 1

0,5

1

-10

1

0

+5

+2

0,5

+1

1

Attendre

1

0

+2

0,5

0,5

-10

Rechercher

Intervention d’un opérateur

Déchargé

Rechercher

Rechercher

Rechercher

Aller à la station

Attendre

A bloc

Presque vide

**Dans l’état à bloc :**

* Rechercher
  + 50 % de chance de rester à bloc avec une récompense de +5 car plus le robot va rechercher, plus il va trouver de cannettes.
  + 50 % de chance de tomber à l’état presque vide avec +2 en récompense.
* Attendre
  + 100% de chance de rester à bloc car le robot ne va pas se décharger. 0 en récompense car le robot peut quand même récupérer des cannettes mais on en veut le maximum.

**Dans l’état déchargé :**

* Intervention d’un opérateur
  + 100% de chance de retourner à bloc, mais une récompense de -10 car on ne veut pas qu’un opérateur intervienne.

**Dans l’état presque vide :**

* Rechercher
  + 50% de chance de rester à presque vide avec une récompense de +2 car on utilise au maximum la batterie.
  + 50% de chance de tomber à déchargé, ce qui n’est pas souhaité, car nous ne voulons pas faire intervenir d’opérateur, donc une récompense de -10.
* Attendre
  + 100% de chance de rester à l’état presque vide, mais sans récompense car le robot doit ramasser le maximum de cannettes.
* Aller à la station
  + 100% de retourner à bloc avec une récompense de +1 car cela évite l’état « déchargé ».