Algorithmes itératifs pour les processus de décision Markovien

1 Programmation dynamique à horizon infini

Dans cette première partie, vous devez programmer, dans le langage de votre choix, l'algorithme de programmation dynamique avec récursivité inverse afin de déterminer la politique optimale dans un problème particulier. Ce problème est la recherche du chemin le plus coûteux dans un arbre de profondeur T comme illustré sur la figure suivante.

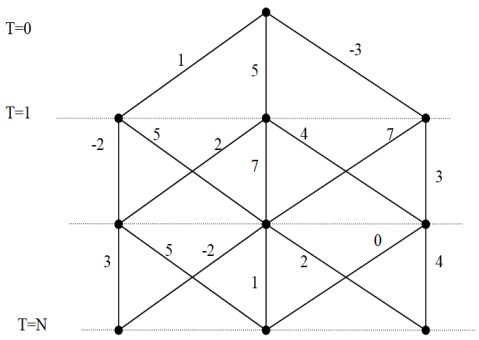

FIGURE 1 – Exemple d'arbre pondéré avec $T = 3$.

Question 1 *Vous devez tout d'abord écrire une structure de données d'arbre pondéré comme celui de la figure ci-dessus.*

Question 2 *Un premier programme déterminera par la méthode de récursivité inverse la fonction de valeur dans chaque état du graphe pour chaque niveau de l'arbre.*

Question 3 *Un second programme calculera la politique optimale et donc le chemin le plus long depuis la racine ($k = 0$) jusqu'aux feuilles ($k = T$).*

2 Programmation dynamique à horizon infini

L'objectif de cette seconde partie est de programmer, dans le langage de votre choix, les algorithmes itératifs vus en cours pour résoudre les processus de décision Markovien à horizon de temps infini. Ces algorithmes seront testés sur un jeu de déplacements aléatoires d'un robot sur une surface de jeu avec obstacles. Chaque case du plateau correspond à un état a. On peut supposer que la probabilité d'aller dans la direction demandée est toujours de 80%, et les autres directions possibles se répartissent uniformément le reste de la probabilité.

Votre programme devra définir entre autres les caractéristiques suivantes du MDP :

— taille du plateau de jeu ($n \times m$ avec n lignes et m colonnes),
— position de départ du robot (couple $(x_0, y_0) = (1, 1)$),
— valeur de la récompense dans chaque état $s = (x, y)$ avec $x \in \{1, 2, \ldots, n\}$ et $y \in \{1, 2, 3, \ldots, m\}$,
— probabilités de transition pour chaque action et couple d'états, i.e. $P(s'|s, a)$, en supposant une erreur. Pour chaque action demandée, il y a une probabilité de 0.2 que le robot fasse une autre action (déplacement) avec un angle de 90 degrés. Cette probabilité de déviation de l'action demandée est répartie entre les actions possibles avec un angle de 90 degrés.
— paramètre d'escompte $\gamma \in [0, 1]$.
— la récompense dans chaque état est nulle sauf pour l'état $(2, 4)$ (resp. $(3, 4)$) qui est de -1 (resp. $+1$).

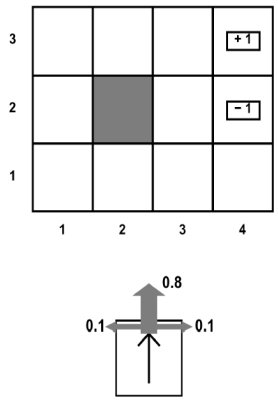

FIGURE 2 – Exemple de plateau de jeu de taille 3×4 et pour chaque action, par exemple vers le haut, la probabilité de monter est 0.8, d'aller à gauche 0.1 et à droite 0.1.

Question 4 *Un premier programme correspond à l'algorithme de programmation dynamique par itération de la valeur.*

Question 5 *Un second programme correspond à la détermination de la politique optimale par itération de la politique.*

Vous devez rendre un rapport sur votre travail par email à yezekael.hayel@univ-avignon.fr pour le 31 mars.