DEVOIR PROCESSUS DÉCISIONNELS AVANCÉS

Abdel-Illah Mouaddib
Master 2, Décision et Optimisation,
Université de Caen Normandie

À RENDRE LE 31 JANVIER 2025

1 Familiarisation avec la bibliothèque MADP Toolbox

La bibliothèque sur les algorithmes DECPOMDP et les benchmarks contient tous les outils pour le développement et l'évaluation des algorithmes autour du DEC_POMDP et POSG. Toutes les explications pour le fonctionnement, le chargement et l'installation de cette bibliothèque se trouvent à
`https://www.fransoliehoek.net/fb/index.php?fuseaction=software.madpinfo`

1. Télécharger la bibliothèque MADP : MADP toolbox v0.4.1 (.tar.gz) et installer l'outil (./configure , make, make install). Détails sont donnés sur le site.
2. Charger le bechnmark de la gridmeeting 2×2 à partir du lien :
   `http://rbr.cs.umass.edu/camato/decpomdp/down/GridSmall.posg.dat`
   dont la description est comme suit :
   - Espace d'états : 16 états joints = $\{0, \ldots, 15\}$. Un état joint correspond à la présence des deux agents sur les cellules de la grille numérotées ligne par ligne $\{0, 1, 2, 3\}$. Ainsi l'état joint 0 correspond à l'état (0,0) signifiant que les agents 1 et 2 partage la cellule 0 de la grille. L'état joint 1, correspond à (0,1) signifiant que l'agent 1 est sur la cellule 0 de la grille et l'agent 2 sur la cellule 1 de la grille et ainsi de suite.
   - Espace d'actions : 5 actions par agent = $\{1, 2, 3, 4, 5\}$ correspondant respectivement à UP, DOWN, LEFT, RIGHT, STAY. Espace d'observations : deux observations par agent = $\{1, 2\}$ signifiant que un mur est à gauche ou à droite.
   - La fonction transition est définie sous la forme *T(état joint départ, action agent 1, action agent 2, état joint d'arrivée).*
   - La fonction d'observation est définie sous la forme *O(observation agent 1, observation agent 2, état joint, action agent 1, action agent 2).*

— La fonction de récompense est définie comme suit : les états joints 0, 5, 10, 15 correspondants que les 2 agents partagent la même cellule ont une récompense de +1 et le reste 0.

3. Lancer l'exemple avec l'algorithme exhaustif et l'algorithme JESP et donner les politiques produites.

2 GridMeeting

1. On reprend le benchmark de la section ci-dessous et on souhaite le résoudre avec une approche naıve qui consiste ce que l'agent 2 suit une politique qui lui fait appliquer l'action STAY tout le temps et que l'agent 1. calcule une BEST_RESPONSE pour le rejoindre. En gardant le même Benchmark, Lancer un algorithme BEST_RESPONSE avec la politique de l'agent 2 connue et donner la politique de l'agent 1.
2. Comparer cette politique avec une politique d'un MDP qui permet à l'agent 1 d'aller à la position de l'agent 2.
3. A votre avis, sous quelle condition les deux politiques (MDP et BEST_RESPONSE) sont identiques
4. En reprenant le benchmark de la section ci-dessous, on souhaite que les deux agents ne se rencontrent pas. Proposer une modification du modèle pour tenir compte de cet objectif des agents. Relancer les algorithmes exhaustif et JESP et donner les politiques produites.

3 Leader-Suiveur

Dans cette exercice, on vous propose de créer un benchmark pour que deux robots se dirigent vers une destination en file indienne avec un leader et un suiveur. On peut s'inspirer du modèle GridMeeting mais pour une grille 4×4, les robots ont les mêmes actions : UP, DOWN, LEFT, RIGHT, STAY, les états joints sont au nombre de 16^2 sur les mêmes principes que sur l'exemple GridMeeting et l'espace d'observation sont de deux par agents (mur gauche, mur droite).

1. On vous demande de fournir un modèle de ce problème en supposant la transition et l'observation indépendantes avec les principes suivants :
   — une action atteint sa cellule cible à 0.9 et ls cellules voisines gauche et droite avec 0.05, 0.05.
   — La fonction d'observation est fiable à 0.95 pour voir un mur quand il existe et peut voir des murs qui n'existent pas à 0.1. Transcrire ces données dans la fonction d'observation de chaque agent.
   — la fonction de récompense est calculée en fonction de la moyenne des distances qui séparent les deux robots de la destination. Quand les deux robots sur la cellule de destination la récompense est maximale.
2. Pour résoudre ce problème, on va utiliser deux méthodes :

- **Méthode 1 :** Utiliser l'algorithme JESP.
- **Méthode 2 :** On utilise l'algorithme BEST_Response de la manière suivante :
  - choisir un leader selon une heuristique que vous proposez.
  - calculer la politique optimale du Leader π_L par l'algorithme VI
  - calculer la politique du suiveur π_s comme une Best_response à la politique du Leader π_L.