

Projet Planification Multi-agents

Collecte multi-agents de trésors

Master DAI 2023-2024

1 Instructions pour le rendu du projet

Le projet devra être rendu par mail pour le **20 février 2024 au plus tard**.

Le projet est à réaliser en Python à partir des fichiers fournis sur Moodle.

Vous devez rendre :

- Le code de votre projet commenté.
- Un rapport au format PDF décrivant la structure du programme et vos choix d'implémentation (algorithmes, protocoles, etc). Vous présenterez également les résultats de l'exécution de votre programme sur différentes instances pertinentes du problème.

Le projet peut être réalisé en binôme ou de manière individuelle.

2 Problème étudié : collecte de trésors multi-agents

- On considère des agents devant ramasser la totalité des trésors répartis dans un environnement.
- L'environnement est représenté par une grille. Un agent peut se déplacer d'une case à une autre si elles sont adjacentes (les déplacements en diagonale sont permis). Deux agents ne peuvent pas se trouver sur la même case.
- Des trésors sont répartis dans l'environnement. Il existe 2 types de trésors : les pierres précieuses et les pièces d'or. Ces trésors sont contenus dans des coffres qui doivent être déverrouillés avant que les agents puissent ramasser les trésors (dès qu'un coffre est déverrouillé, un agent peut ramasser le trésor correspondant sans attendre que tous les coffres soient déverrouillés). Pour ramasser un trésor ou déverrouiller un coffre, un agent ayant la bonne compétence doit se déplacer sur la case du trésor et effectuer l'action voulue.
- Les agents ont 3 types différents : les agents ramassant des pièces d'or, les agents ramassant des pierres précieuses et les agents ouvrant les coffres. Un agent ramassant des pièces d'or ne peut pas ramasser des pierres précieuses et réciproquement. Un agent ouvrant les coffres ne peut ramasser aucun trésor.
- Chaque trésor peut être ramassé par un seul agent, à condition que le coffre ait été déverrouillé. Une fois ramassé, un trésor doit être déposé

dans un point de collecte identifié. Seuls les trésors déposés dans ce point de collecte sont comptabilisés.

- Chaque agent dispose d'une capacité de sac à dos qui correspond à la quantité maximum de trésor qu'il peut ramasser. Si la quantité à ramasser est plus grande que la place restante dans le sac à dos, la partie non ramassée du trésor est définitivement perdue. Les agents peuvent toutefois décharger leur sac au point de collecte final pour ensuite collecter d'autres trésors.

La figure 1 présente un environnement de 12 cases sur 12 cases avec 7 agents : 2 agents ouvrant les coffres, 3 agents ramassant les pièces d'or et 2 agents ramassant les pierres précieuses. Les pierres précieuses sont marquées en rouge et les pièces d'or en jaune. Le point de collecte (ou appelé aussi zone de dépôt) est situé au milieu de la première colonne. La valeur précisée sur chaque trésor correspond à la quantité de trésor à ramasser.

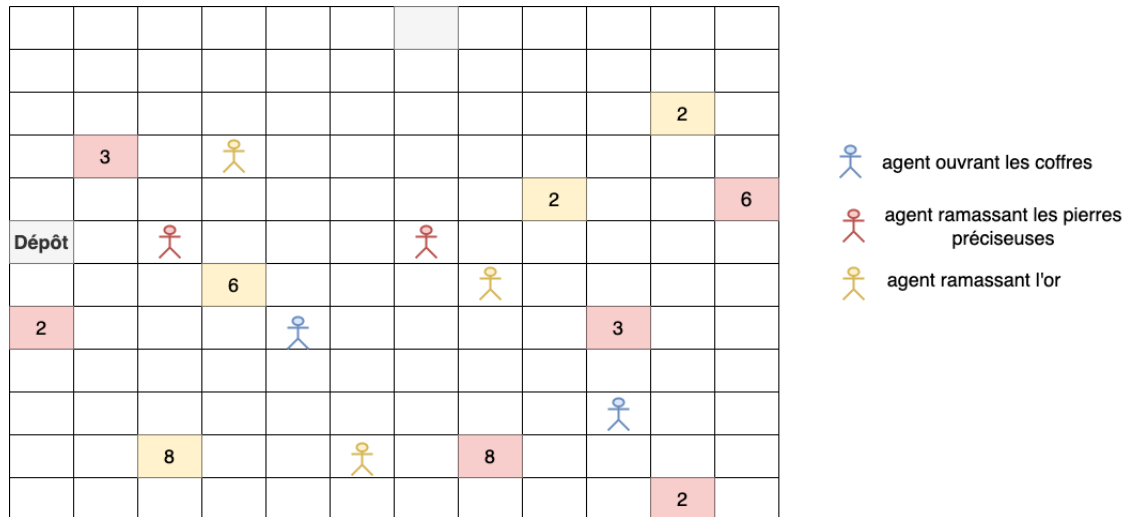


FIGURE 1 – Exemple d'environnement

La description d'une instance d'un problème se fait dans un fichier texte suivant le format suivant (à noter que les agents qui ouvrent les coffres n'ont pas de capacité sac à dos) :

```
#taille env
12 12
# position depot
5 0
# tresors tres:type:posX:posY:value
tres:or:2:10:2
tres:pierres:3:1:3
tres:or:4:8:2
tres:pierres:4:11:6
tres:or:6:3:6
tres:pierres:7:0:2
```

```
tres:pierres:7:9:3
tres:or:10:2:8
tres::10:7:8
tres:or:11:10:2
#agents AG:type:posX:posY:backpack
AG:ouvr:7:4
AG:ouvr:9:9
AG:pierres:5:2:9
AG:pierres:5:6:15
AG:or:6:7:6
AG:or:10:5:17
```

3 Objectif du projet

L'objectif du projet est de développer une méthode de planification **distribuée** permettant aux agents de coordonner la collecte des trésors de sorte à maximiser la quantité de trésors déposée au point de collecte.

Pour se faire vous devrez développer des algorithmes et protocoles permettant aux agents :

1. de se répartir les coffres à ouvrir et les trésors à ramasser (en respectant les types des agents et les capacités des sacs à dos),
2. de planifier de manière individuelle leurs actions,
3. de coordonner de manière distribuée les plans individuels afin d'obtenir un plan global efficace. On remarquera que la coordination est en particulier nécessaire pour permettre de répondre aux contraintes de non collision (2 agents ne peuvent être au même moment sur la même case), de précedence entre l'ouverture d'un coffre et le ramassage d'un trésor, et afin de coordonner le ramassage.

La configuration initiale de l'environnement (taille de l'environnement, position de la zone de dépôt, position initiale des agents, des trésors) est connue de tous les agents. En revanche, les agents n'observent pas l'évolution de l'environnement au cours de l'exécution des actions.

Vous devez télécharger le code de base du projet et compléter ce code. Ce code, écrit en Python, propose les classes et méthodes de bases pour créer des agents dans un environnement et faire évoluer ces agents dans l'environnement. Le fichier `Main.py` donne des exemples d'appels des fonctions fournies. Vos modifications se feront essentiellement dans la classe `Main.py` et dans les classes `Agent`. Vous pouvez créer de nouvelles classes. Vous ne devez pas modifier la classe `Environment` ni les méthodes déjà écrites dans la classe `agent`. Vous ne devez pas utiliser de structure de données partagées entre tous les agents. Les agents communiqueront par envois de messages par l'appel à la méthode `send`.

Il est conseillé de créer différentes instances du problème et de tester votre programme sur ces différentes instances. Une instance est chargée à l'aide de la méthode `load` de la classe `Environment`.

4 Allocation des tâches entre agents

1. Proposez et implémentez un protocole d'allocation des tâches (ramassage des trésors et ouverture de coffres) entre les agents, et respectant les capacités des agents. On veillera bien entendu à équilibrer les tâches entre les agents.

5 Planification individuelle et coordination distribuée

1. Étant donnée la liste des tâches d'un agent, mettez en place une méthode de planification des tâches de l'agent lui permettant d'optimiser la réalisation de ses tâches.
2. Mettez en place une méthode de coordination distribuée permettant aux agents d'échanger des informations sur leurs plans individuels et de se coordonner.
3. Permettez aux agents d'exécuter leurs plans. Une interface graphique est possible (points bonus).