

Faculté des Sciences et Ingénierie
Sorbonne Université

Projet - Wumpus Multi-agent

MICHAËL TRAZZI, MICHAËL AIDAN

Sous la supervision d'AURÉLIE BEYNIER, NICOLAS MAUDET et CEDRIC HERPSON

Master 1 Informatique, spécialité ANDROIDE
Année 2017 – 2018, Semestre 2



Table des matières

1	Introduction	2
1.1	Présentation du probleme	2
1.2	Problématiques	2
1.3	Ressources	2
2	Analyse des algorithmes	3
2.1	Exploration	4
2.1.1	Principe	4
2.1.2	Avantages/inconvénients	4
2.1.3	Complexité	4
2.2	Communication	4
2.2.1	Principe	4
2.2.2	Avantages/inconvénients	4
2.2.3	Complexité	4
2.3	Interblocages	4
2.3.1	Principe	4
2.3.2	Avantages/inconvénients	4
2.3.3	Complexité	4
2.4	Ramassage des Trésors	4
2.4.1	Principe	4
2.4.2	Avantages/inconvénients	4
2.4.3	Complexité	4
2.5	Coordination	4
2.5.1	Principe	4
2.5.2	Avantages/inconvénients	4
2.5.3	Complexité	4
2.6	Migration	4
2.6.1	Principe	4
2.6.2	Avantages/inconvénients	4
2.6.3	Complexité	4
3	Conclusion	5
3.1	Synthese	5
3.2	Regard critique sur notre travail	5
3.3	Extensions et améliorations possibles	5

1 Introduction

Pour ce projet nous avons été amenés à implémenter un Système Multi-Agent pour résoudre un problème en apparence simple : la collecte de trésors sur un graphe. Cependant, l'existence d'une perception limitée des agents, et la présence d'un terrifiant « Wumpus » sur la carte a rendu difficile l'implémentation d'une solution satisfaisante.

1.1 Présentation du probleme

Il y a 4 types d'agents : les Explorateurs, pouvant uniquement se déplacer, les Collecteurs, capables de ramasser les trésors de leur type jusqu'à remplir leur sac, les agents Silo, pouvant accumuler sans limite les trésors ramassés par les agents Collecteur, et l'agent « Wumpus », se mouvant aléatoirement sur la carte en déplaçant les trésors.

Les seuls agents dont nous pouvons modifier le comportement sont les agents Explorateurs, Collecteurs et Silo. Ces agents vivent dans un environnement **JADE**, environnement de développement pour des systèmes multi-agents en **Java**. Ils exécutent des comportements (ou *behavior*) séquentiellement, dans un ordre pré-déterminé, à chaque fois qu'un processus les réveille.

En Intelligence Artificielle, un agent est n'importe quel objet percevant son environnement par des *senseurs*, et agissant sur le monde à travers des *effecteurs*. Le Wumpus n'ayant pas de comportement complexe (il ne prend pas de décision mais se contente de faire des mouvements aléatoires), nous désignerons par *agent* les trois autres agents (Explorateurs, Collecteurs et Silo). Ces trois agents (qu'on a été amené à implémenter) disposaient de senseurs et effecteurs relativement précaires.

- **Senseurs**

- Observation Un agent peut exécuter la méthode **observe**, qui retourne une liste d'attributs des noeuds adjacents au noeud où se trouve l'agent. L'agent sait donc quels noeuds sont adjacents, et si ces noeuds contiennent des trésors. Ils n'a pas *a priori* connaissance de la présence d'un autre agent sur un de ces noeuds.

- Réception de message.

- **Effecteurs**

- Mouvement

- Collecte de trésor

- Transmission de trésor

- Emission de message

1.2 Problématiques

Où on explique les problématiques résultantes du probleme présenté ci-dessus.

1.3 Ressources

Notre implémentation est disponible sur **github** à l'adresse suivante : <https://github.com/mtrazzi/Hunt-The-Wumpus>. De plus, nous avons rédigé un article, publié en ligne, expliquant en détail les difficultés liées aux Systèmes Multi-agents, et en particulier pourquoi implémenter des protocoles Multi-Agents pouvait s'avérer bien plus complexe que simplement faire de l'apprentissage machine supervisé. Cet article de vulgarisation est hébergé sur **Medium** : <https://hackernoon.com/why-coding-multi-agent-systems-is-hard-2064e93e29bb>.

2 Analyse des algorithmes

Introduction générale des différents algorithmes.

2.1 Exploration

2.1.1 Principe

2.1.2 Avantages/inconvénients

2.1.2.1 Forces

2.1.2.2 Limites

2.1.3 Complexité

2.1.3.1 Critere d'arret

2.1.3.2 Temps

2.1.3.3 Mémoire

2.1.3.4 Communication

2.1.3.5 Optimalité

2.2 Communication

2.2.1 Principe

2.2.2 Avantages/inconvénients

2.2.2.1 Forces

2.2.2.2 Limites

2.2.3 Complexité

2.2.3.1 Critere d'arret

2.2.3.2 Temps

2.2.3.3 Mémoire

2.2.3.4 Communication

2.2.3.5 Optimalité

2.3 Interblocages

2.3.1 Principe

2.3.2 Avantages/inconvénients

2.3.2.1 Forces

2.3.2.2 Limites

2.3.3 Complexité

2.3.3.1 Critere d'arret

2.3.3.2 Temps

2.3.3.3 Mémoire

3 Conclusion

3.1 Synthèse

3.2 Regard critique sur notre travail

3.3 Extensions et améliorations possibles