CARVALHO Lorenzo

PROJET FOSYMA

Sommaire :

Introduction

Dans ce projet, l'objectif est de créer une équipe d'agent qui sache se coordonner pour explorer un labyrinthe et récupérer des ressources, tout en évitant les pièges de ce labyrinthe (wumpus et fosse).

Pour ce faire, les agents vont devoir coopérer, tout d'abord en réussissant à explorer chaque nœud de l'environnement dans lequel ils se trouvent, puis en allant chercher un maximum de ressources dans un minimum de temps, tout ça en essayant de distribuer les ressources entre agents de la manière la plus équitable possible.

Fonctionnement des agents

Ici, on utilisera un seul type d'agent, qui adoptera plusieurs comportements au fil du temps. Dans un premier temps il va vouloir explorer la carte, il le fera en explorant les nœuds non explorés les plus proches. Après avoir découvert toute la carte, il aura donc accès à la localisation de toutes les ressources, et il déterminera son parcours pour récupérer des ressources. Nous allons détailler ces étapes ci-dessous, en ajoutant les communications entre agents.

I- Exploration et communication

Lors de l'exploration, chaque agent envoie à intervalle régulier un message destiné à n'importe quel agent qui serait assez proche pour le recevoir. Ce message contient assez peu d'informations pour ne pas encombrer le réseau, mais s'il reçoit une réponse à ce message, alors il partagera plus d'informations : sa connaissance actuelle de la carte, l'emplacement des trésors connus, ainsi que le lieu de rendez-vous d'après exploration s'il le connait. Après avoir reçu les mêmes informations de la part de l'agent avec lequel il communique, il fusionnera les informations et continuera son exploration. Lorsqu'un agent a fini son exploration, soit il a récupéré le point de rendez-vous et il s'y rend, soit il va bouger aléatoirement en cherchant d'autres agents qui pourraient lui donner.

II- Collecte

Lorsqu'un agent a fini son exploration, il va chercher à récoler des ressources. Soit l'agent est le leader et sera celui qui distribue les points de récolte aux autres agents, soir l'agent attendra qu'on lui attribue ses points de récolte. Le but principal étant l'équité des ressources, le leader sait exactement ce que chacun doit récolter car il connait le nombre de ressource disponible sur la carte ainsi que le nombre d'agents. On prendra en compte le fait qu'il y ait différent type de ressource dans le calcul de l'équité, et on essaiera de

minimiser la différence entre le butin d'agents qui s'occupent de 2 ressources différentes.

III- Blocages

Que ce soit en exploration ou en collecte, les agents feront face à divers obstacles, le wumpus, les fosses, et même simplement le fait qu'un autre agent soit la peuvent bloquer son chemin ou le tuer. Pour éviter ça, chaque agent possède un système de déblocage. Lorsqu'il rencontre un obstacle sur sa route, c'est-à-dire qu'il n'a pas pu rejoindre le nœud qu'il aurait dû rejoindre d'après son path, il fait un mouvement aléatoire. Ce mouvement aléatoire peut également correspondre au fait de rester sur place, pour répondre au cas où il serait coincé sans aucune autre solution. Le blocage n'est pas le seul problème à résoudre, car dans le cas où il rencontrerait un wumpus ou tomberait dans une fosse, un agent pourrait tout simplement disparaitre, ce qu'on cherche évidemment à éviter. Pour ce faire, on sait que lorsqu'un agent se tient sur une case assez proche d'un de ces 2 éléments problématiques il peut le savoir. Donc pour éviter de passer par ces chemins et pour ne pas bloquer l'exploration et la collecte de l'agent, on représente ces nœuds comme des nœuds fermés, comme si c'était des culs-de-sac. Mais il reste un problème, car il est possible que l'agent se retrouve dans un de ces culs-de-sac s'il se retrouve coincé et qu'il fait un mouvement aléatoire. Dans ce cas-là on ne veut pas que l'agent enregistre les nœuds, même s'il les ferme de suite, car il pourrait s'agir d'une fosse. On va donc sauvegarder chacun de ces nœuds, qu'on appellera « nœuds dangereux », et à chaque fois qu'un agent arrive sur un nouveau nœud il devra vérifier s'il s'agit ou non d'un nœud dangereux.

Equité

Le système d'équité est mis en place à la fin de l'exploration, lorsque tous les agents ont une connaissance complète de la carte et de ses ressources. Les agents se retrouvent alors autour d'un point de la carte, à ce moment-là, le leader calcule les combinaisons de trésors les plus équitables possibles, en prenant en compte le fait qu'il y ait 2 types de trésors différents. Pour ce faire il va tester des combinaisons de 2 groupes d'agents en divisant le nombre total de chaque ressource par le nombre d'agent d'un des groupes. Quand le nombre de ressource par agent sera le plus équitable, le leader enverra les informations aux agents les plus proches, qui les transmettront aux agents les plus proches etc…

Complexité et terminaison

Pour l'exploration, on utilise l'algorithme de l'exploration des frontières, la seule différence avec l'algorithme de base étant que normalement un agent va forcément vers la frontière la plus proche, ici, il peut aller vers une frontière plus éloignée s'il croise un agent et qu'ils se répartissent les frontières restantes.

Pour l'équité, il n'y a qu'un seul agent qui s'en occupera. Parmi les agents, un sera nommé leader, lorsqu'il arrivera au point de rendez-vous il attendra un certain temps, où il en profitera pour récupérer les informations des agents qui arrivent les uns après les autres. Ensuite il calculera la distribution optimale des ressources. Il va diviser les agents en 2 groupes et diviser le nombre de chaque ressource par le nombre d'agent d'un groupe, et garder la configuration la plus équitable. Ensuite il attribuera à chaque agent les coordonnées des trésors qu'il doit aller récupérer.

Pour la récolte, chaque agent récupère une liste de trésors qui lui est attribuée. Ensuite ils vont calculer l'ordre dans lequel ils vont aller les chercher, ils iront à

chaque fois vers le trésor le plus proche d'eux, en partant de leur position de départ pour le 1er, puis du dernier trésor ramassé pour les autres. Quand ils auront ramassé tous les trésors, ils s'effaceront eux-mêmes.

Par rapport à la terminaison de ces phases en général, il y a beaucoup de paramètres qui font qu'il est compliqué d'avoir une terminaison « propre ». Par exemple si un agent est bloqué lors de l'exploration, il peut y avoir plusieurs raisons à cela, que ce soit à cause d'agents ou de fosses par exemple. Sauf que dans un cas, les agents peuvent se débloquer et le libérer, alors que dans l'autre il est condamné à rester bloquer jusqu'à la fin du processus.

Pour assurer la terminaison il y a donc des compteurs dans chaque étape : exploration, regroupement et collecte.

Si au bout d'un certain nombre de pas, l'étape n'est toujours pas réalisée pour l'agent, alors il passera à la suivante. En sachant que seul le leader peut passer du regroupement à la récolte, s'il n'est pas là alors un agent prendra sa place.

Ce nombre doit être assez élevé pour permettre aux agents qui mettent un peu plus de temps à revenir, mais il y a également une terminaison plus rapide pour les agents immobiles. Si un agent reste immobile un certain nombre de temps, alors il se supprimera également.

Synthèse

En conclusion, les agents partent de n'importe où et arrivent à se coordonner sans se bloquer, arrivent à se regrouper et ensuite à aller chercher des ressources avant de s'effacer eux-mêmes. Les bases sont donc fonctionnelles.

Dans un environnement statique tout fonctionne comme prévu, mais dans un environnement où l'on rajoute des éléments qui peuvent bouger, il existe des situations ou ce programme peut être en difficulté et ne pas fonctionner comme il le devrait.

Critique et améliorations :

L'algorithme de déblocage marche bien, dans une map vide les agents se coordonnent parfaitement et il n'y a pas de blocage. Même chose dans une map vide avec des objets statiques (fosses et wumpus immobile), par contre il peut y avoir des problèmes avec un wumpus qui bouge, car comme les agents ferment le nœud où ils sentent le wumpus, il est possible qu'il soit par exemple sur un nœud qui soit le seul accès à une partie de la map, et que les agents considèrent que ce nœud est fermé et donc que la map s'arrête la. Même chose s'il se tient sur une ressource ou s'il en déplace une, les agents ayant déjà construit leur map à cet endroit ne verront pas la ressource.

Pour résoudre ce problème il faudrait donc un moyen que chaque agent ait une version optimale de la carte.

Un autre problème serait l'optimalité du chemin parcouru pour récupérer les ressources. En effet, les ressources ne sont pas distribuées aléatoirement car il y a un principe d'équité entre agents, mais elles ne sont pas non plus distribuées de manière à minimiser les trajets de chaque agent. Au niveau des ressources on peut avoir le même problème que le premier cité, à savoir que comme les agents ne rafraichissent pas la map après des changements, si le wumpus a modifié l'emplacement de l'or, alors les agents ne pourront pas le ramasser.

Et enfin je n’ai pas implémenté de comportement pour gêner le wumpus ou une équipe adverse d’agents. Lorsque les agents ont fini leur exploration ils vont essayer de se retrouver à un endroit puis ils seront immobiles, il y a surement quelque chose à améliorer pour optimiser leurs mouvements.