Polytech Lyon

TP 2 : Tri sélectif

Système multi-agents

Helloïs BARBOSA
Thomas BECHET

Enseignant : Mme. HASSAS

13 mars 2022

Sommaire

1 Architecture

Notre architecture prend la forme d'un environnement dans lequel on trouve des objets (que l'on va nommer ici des fruits pour ne pas confondre avec la class Object de Java), des cases et des agents. Chaque fruit a un type (représenté par sa classe) et chaque case peut posséder un fruit et un ou plusieurs agents. Les agents peuvent récupérer et déposer des fruits depuis les cases. L'environnement est chargé de vérifier la validité des opérations effectuées par les agents (déplacement, dépôt et récupération) puis d'appliquer ces modifications.

La logique d'un agent se base sur le patterne d'un automate fini à plusieurs états. Les états possibles sont les suivants :

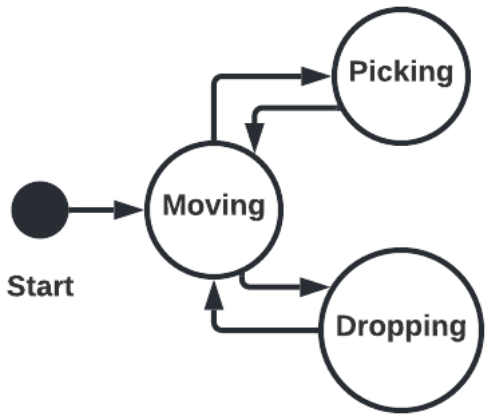

Figure 1 – Diagramme d'état de l'automate

Une mise à jour de l'environnement suit les étapes suivantes : tirage aléatoire d'un agent, perception de l'agent, action de l'agent. Pour faciliter le développement et le débogage du programme, nous avons utilisé une interface graphique pour visualiser les déplacements des agents et la position des fruits.

2 Tri sélectif simple

Pour cette première partie, nous avons choisi deux types de fruits : les bananes (en jaune) et les pommes (en rouge). Les agents sont représentés en bleu. La taille du plateau est de 50 par 50. En utilisant les paramètres initiaux indiqués dans le sujet (200 bananes, 200 pommes et 20 agents, KP = 0.1, KM = 0.3 et 10 pour la taille de la mémoire), on obtient les résultats suivants.

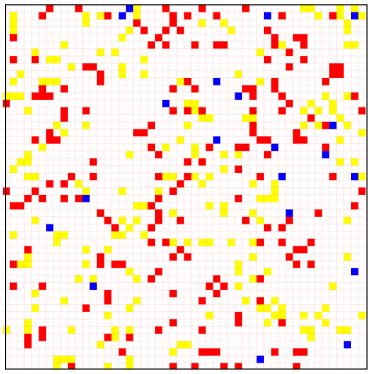

FIGURE 2 – Etat initial de l'algorithme : les fruits ne sont pas triés

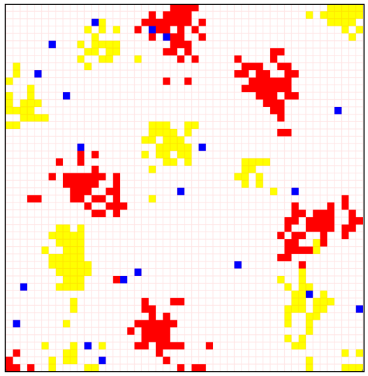

FIGURE 3 – Etat de l'algorithme à l'itération 1 000 000 : les agents formes des petits clusters

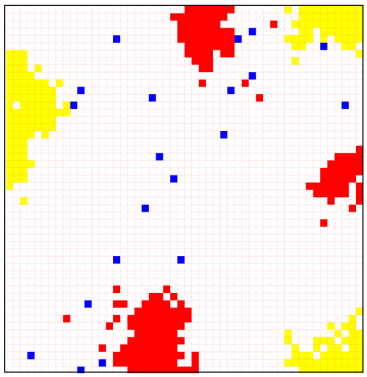

FIGURE 4 – Etat de l'algorithme à l'itération 10 000 000 : les agents ont trié les fruits

Après l'itération 10 000 000, le tri semble avoir convergé, nous n'observons plus de modifications sur les groupes de fruits déjà formés.

Nous avons ensuite implémenté la possibilité qu'un agent laisse tomber un fruit ou le prenne par erreur en faisant en sorte que sa perception soit modifiée. Pour cela, nous avons créé la classe AgentDropPickError qui surcharge la méthode de calcul de la fréquence des fruits dans sa mémoire. Il est ainsi possible de modifier la formule et d'y insérer un taux d'erreur.

3 Tri sélectif complexe

Dans une seconde partie, nous avons ajouté des fruits nécessitant plusieurs agents pour être transporté, dans notre cas, il s'agit des Kiwi (couleur verte) qui ont besoin d'exactement deux agents pour être déplacé.

Pour que les agents se retrouvent sur la case où se trouve un kiwi nous avons donné la possibilité à un agent d'émettre un signal lorsqu'il est sur une case où se trouve un fruit nécessitant plusieurs agents pour le déplacer. Ce dernier sera alors en attente pendant un certain laps de temps avant d'abandonner si aucun autre agent ne lui vient en aide. Nous avons donc modifié le diagramme d'états de l'agent en ajoutant l'état "EMIT".

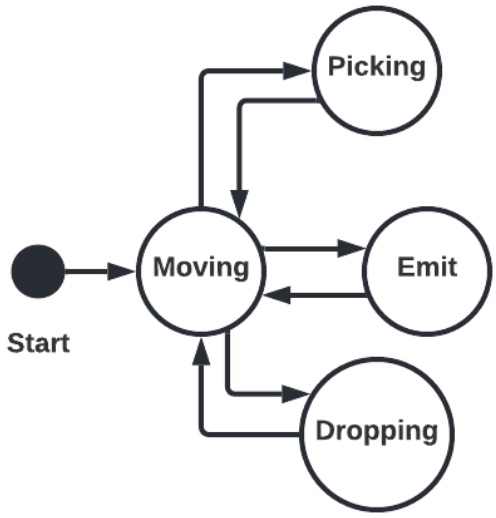

FIGURE 5 – Diagramme d'état de l'automate complexe avec la partie émission d'un signal

Lors de l'émission d'un signal, les cases aux alentours voient leur signal augmenté. À chaque mise à jour de l'environnement, on vient réduire le signal avec la formule suivante :

$$S = S - (S * r) \tag{1}$$

Pour gérer correctement le déplacement des groupes, nous avons dû modifier les vérifications réalisées par l'environnement lors du déplacement d'un agent.

Finalement, pour influencer le déplacement d'un agent vers les cases possédant un signal plus élevé, l'agent va réaliser un tirage avec poids des cases dans son champ de vision. Le poids d'une case est calculé de la façon suivante :

$$W = 1 + (S/S_{max}) \tag{2}$$

Ainsi, une case ayant un signal de zéro aura un poids de 1. Ce tirage avec poids nécessite plus de calculs lors de la décision et cela ralenti donc beaucoup la simulation. Pour réduire le temps de simulation, nous avons réduit la taille de l'environnement de 50x50 à 30x30.

Visuellement, la valeur du signal pour une case est représentée par la couleur de sa bordure. Plus elle est rouge, plus son signal est important. Deux agents formant un groupe sont représentés en violet.

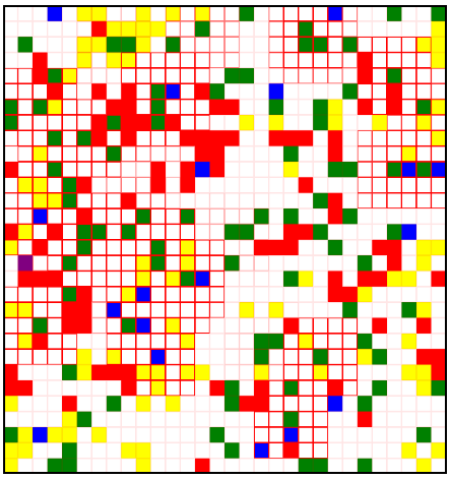

FIGURE 6 – Etat initial de l'algorithme avec émission du signal

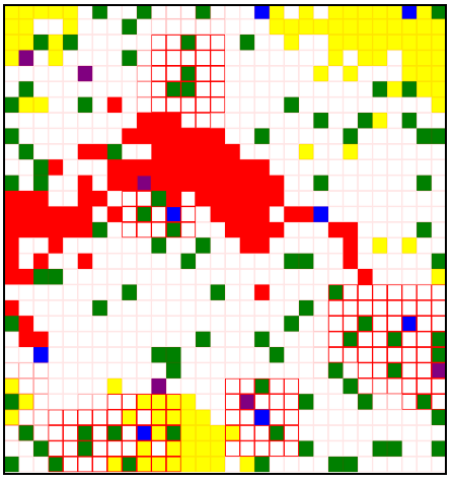

FIGURE 7 – Etat à 10 000 000 itérations de l'algorithme avec émission du signal

On observe que les agents arrivent toujours à trier les fruits pouvant être déplacé seul, mais les fruits nécessitant plusieurs agents (ici les Kiwi verts) ne sont que très peu regroupés bien qu'ils soient bien déplacés. Nous n'avons malheureusement pas trouvé de solutions à ce problème, faute de temps.