Système Multi-agent

Tri collectif partie 2

Marion Vertessen et Dane Badiel

Introduction

Ce rapport se rapporte à un travail pratique sur le tri collectif réalisé dans le cadre de nos études à l'université Lyon 1. Il consiste à détailler la simulation d'un tri collectif avec des agents réactifs comme évoqué dans l'article de J.L Denebourg & al. dans *The Dynacis of Collective sorting Robot-Like Ant and Ant-Like Robot.* Pour cela, on possède trois types d'objets à trier (les objets A, les objets B et les objets C) qui sont répartis aléatoirement en quantité n_a, n_b et n_c sur une grille de taille $N \times M$. Des agents déplacent les objets pour les trier en respectant les caractéristiques suivantes :

- Ils se déplacent aléatoirement dans 8 directions d'un pas $i \geq 1$.
- Ils prennent un objet avec une probabilité : $P_{prise} = \left(\frac{k^+}{k^+ + f}\right)^2$ avec k^+ une constante.
- Ils déposent un objet avec une probabilité : $P_{dépôt} = \left(\frac{f}{k^- + f}\right)^2$ avec k^- une constante.
- On définit une mémoire à court terme des objets que l'agent a déjà rencontré sur les derniers pas pour calculer la valeur de f dans les formules précédentes. De fait, f correspond à la proportion d'objets de même type dans l'environnement immédiat.
- Les objets C nécessitent la collaboration de deux robots afin d'être portés.

Dans un premier temps, on fixe ces différents paramètres selon la *figure 1* ci-dessous :

Taille de la grille	Pas de déplacement	Nombre d'agents	k^+	k^-	Nombre d'objets de type A et B	Nombre d'objets de type C	Taille de la mémoire
50×50	$i = 1$	20	$0,1$	0.3	200	50	10

figure 1 : Tableau des paramètres utilisés pour modéliser le tri

Il est à noter que ce rapport est la suite d'un premier rapport établi en ne considérant que les objets A et les objets B. De fait, celui-ci ne portera pas sur la modélisation principale

utilisée pour le tri sans collaboration. Il se concentrera donc sur la modélisation du système pour les objets C. Cependant, la partie I.Rappel comporte des rappels sur le premier rapport.

I. Rappel

A. Structure du projet

Pour définir la problématique présentée ci-avant, on utilise le langage Python ainsi que la librairie PyQt5 afin de générer l'interface graphique. On définit donc un agent comme une classe. Cet agent est ici considéré comme réactif : il scanne son environnement et agit en conséquence. On peut donc modéliser le comportement d'un agent comme décrit ci-après :

- L'agent scanne son environnement pour déterminer si un objet se trouve sur la case et sa position sur la grille.
- Il met à jour sa mémoire avec l'élément se trouvant sur la case.
- S'il tient un objet, il le dépose sur la case avec une probabilité $P_{dépôt}$ si aucun objet n'est présent sur la case. Sinon, il prend l'objet, s'il y en a un, avec une probabilité P_{prise}.
- L'agent envoie son souhait de se déplacer à l'environnement dans une direction aléatoire.

L'environnement a les caractéristiques suivantes :

- Une grille de taille 50×50 sur laquelle se trouve les objets A et les objets B
- La liste des agents et de leur position
- Il donne la parole à un agent à chaque itération. La parole est donnée aléatoirement.

II. Implémentation

Cette partie se rapporte aux différentes modélisations que nous avons décidés d'utiliser afin de rajouter un comportement collaboratif autour des objets C. La modélisation principale qui sera explicitée dans le reste de cette partie se base sur la propagation de phéromones dans l'environnement.

A. Modélisation

Dans un premier temps, nous avons décidé de tester une modélisation nous permettant de vérifier que les agents portent correctement les objets C.

La perception des agents sont les suivantes :

- La valeur de l'objet qui se trouve actuellement sur leur case qui est soit 0 (si aucun objet se trouve sur la case), soit 1 (si un objet A est présent), soit 2 (si un objet B est présent), soit 3 (si un objet C est présent).
- Sa position sur la grille, en coordonnées de la forme [x,y]
- Une liste comprenant l'ensemble des phéromones présentes dans les cases adjacentes. Dans cette partie, la liste est de taille 8 et ne comprend que les phéromones des cases se trouvant autour de l'agent. Ces cases sont représentées en orange sur la figure 2.

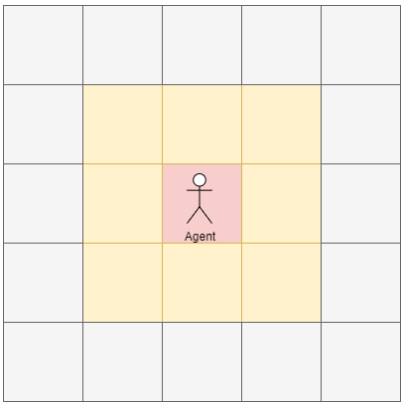

figure 2 : Cases qui sont perçues par l'agent

- La liste des positions accessibles qui est de la même taille que la liste des phéromones. Cette liste permet notamment à l'agent de savoir sur quelle case se trouve chaque phéromone et à se déplacer dans l'environnement.
- Un booléen qui permet au robot de savoir si un autre robot est en train d'attendre de l'aide sur la case.

Les actions de l'agent sur l'environnement :

- Un booléen qui retourne si l'agent est en état d'attente
- Un booléen qui retourne si une collaboration est acceptée
- Un booléen qui retourne si l'agent arrête d'attendre
- La nouvelle position de l'agent
- Un entier qui permet de savoir s'il y a une prise ou un dépôt qui a lieu.

Afin de modéliser cette boucle de perceptions/actions, nous avons utilisé les éléments de modélisation suivants :

1. Si l'agent porte un objet, il le dépose avec une probabilité $p_{dépôt} = \left(\frac{f}{k^{-}+f}\right)^2$ puis se déplace dans une direction aléatoire parmis les positions disponibles
2. Sinon, l'agent choisit ses actions en fonction de son état courant :
    - S'il est déjà en attente, mais que le nombre de tours attendu n'a pas dépassé le nombre de tours d'attente maximale, l'agent continue d'attendre

- Si l'agent est en train d'attendre et qu'il a atteint son nombre maximal de tours d'attente, il arrête d'attendre et se déplace aléatoirement parmi l'une des positions disponibles.
- Si l'agent est sur la même classe qu'un objet de type C, qu'il n'est pas en train d'attendre et qu'un autre robot est déjà en train d'attendre, l'agent rentre en collaboration avec lui pour porter l'objet.
- Si l'objet courant est soit un objet A soit un objet B, l'agent prend l'objet avec une probabilité $p_{prise} = \left(\frac{k^+}{k^+ + f}\right)^2$. Puis il se déplace en fonction des phéromones présentes autour de lui. Il aura plus de chance de se déplacer vers une phéromone avec une plus grande valeur.
- Si aucun objet n'est présent, il se déplace en fonction des phéromones autour de lui.

Afin de modéliser l'appel à l'aide d'un agent, nous avons décidé d'utiliser le système de phéromones. En effet, lorsqu'un robot rencontre un objet C, il émet un signal instancier sous forme d'une information propagée dans le voisinage environnant. Cette phéromone est un flottant compris entre 0 et 1, 0 correspondant à un signal inexistant et 1 pour le signal le plus fort On instancie les variables suivantes :

- $D_s = 2$ correspondant à la distance de diffusion du signal. Sur la figure 3, on retrouve en rouge et en orange l'ensemble des cases qui recevront une phéromone si l'agent appelle à l'aide sur la case centrale.
- $taux_{atténuation} = 0.001$ correspondant au taux d'atténuation du signal au cours du temps. En effet, à chaque fois qu'un agent prend la parole, l'environnement évolue en diminuant les différentes phéromones de cette valeur.
- $diminution_{intensité} = \frac{1}{D_s}$ correspondant à la propagation de la phéromone en fonction de la distance. En effet, l'intensité du signal se réduit au fur et à mesure qu'on s'éloigne de la case où se trouve l'agent de ce taux. On peut retrouver cela sur la figure 3. En effet, la case où se trouve l'agent (en rouge foncé) aura une grande valeur pour la phéromone. Sur les cases rouges claires, on aura une phéromone plus faible et sur les cases oranges une phéromone encore plus faible.

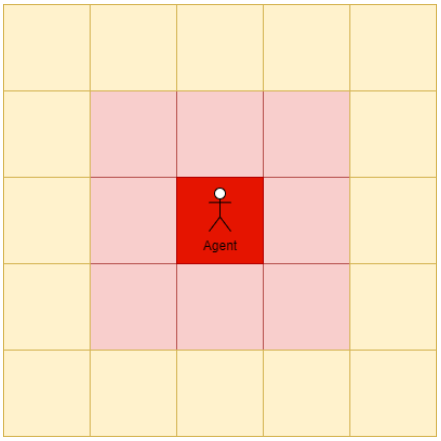

figure 3: Diffusion des phéromones

B. Résultats

Nous avons ensuite testé notre système afin de le valider.

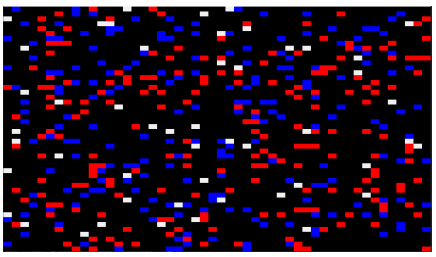

figure 4 : 0 itération

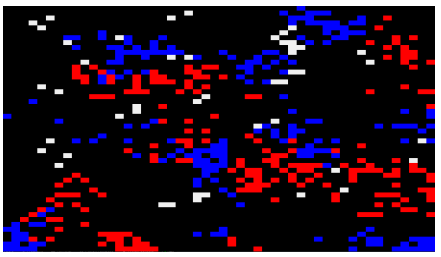

figure 5 : 1 000 000 itérations

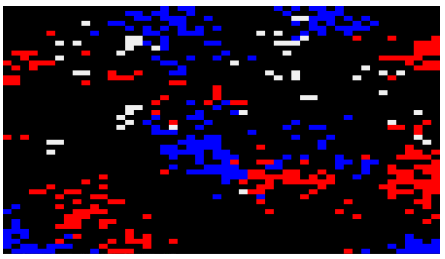

figure 6 : 2 000 000 itérations

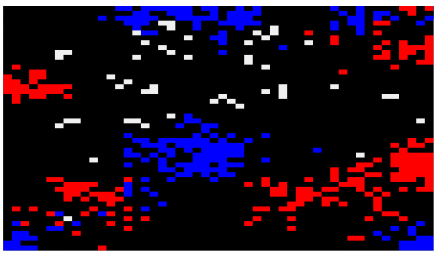

figure 7 : 3 000 000 itérations

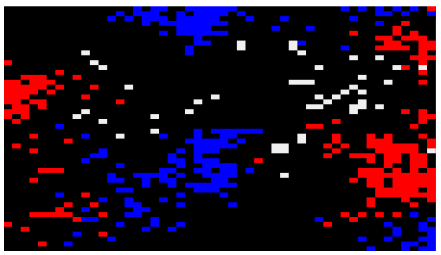

figure 8 : 5 000 000 itérations

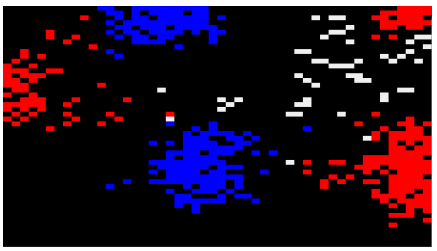

figure 9 : 10 000 000 itérations

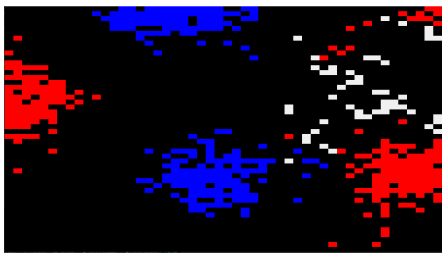

figure 10 : 15 000 000 itérations

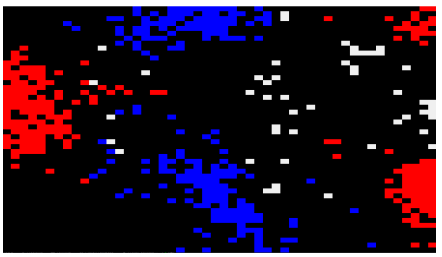

figure 11 : 20 000 000 itérations

On remarque sur les figures 4 à 11 que l'algorithme que nous avons implémenté tri bien les objets A et B. Les objets C sont quand a eu un peu moins bien trié, mais cela reste convenable. En effet, de manière évidente, le tri est plus compliqué à réaliser puisqu'il faut que deux agents collaborent pour déplacer un objet de type C.

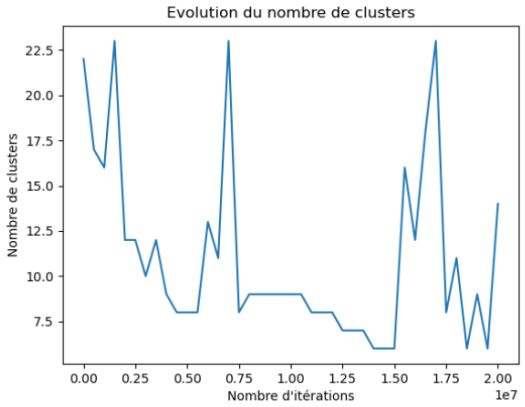

figure 12 : évolution du nombre de clusters en fonction du nombre d'itérations

On remarque sur la figure 12 que le nombre de clusters diminue puis ré-augmente. En étudiant visuellement, cette évolution sur les figures 10 et 12, on remarque que les objets C se divisent après s'être rassemblés.

Cela est dû au fait que dans cette version, lorsqu'un agent trouve un objet C, il appelle à l'aide directement. Cependant, s'il fait cela, il risque de déplacer un objet qui est plutôt bien positionné.

De fait, afin de pallier ce problème, nous avons considéré la probabilité de prise pour les objets C. En effet, un agent trouvant un objet C, appelle à l'aide avec une probabilité pprise.

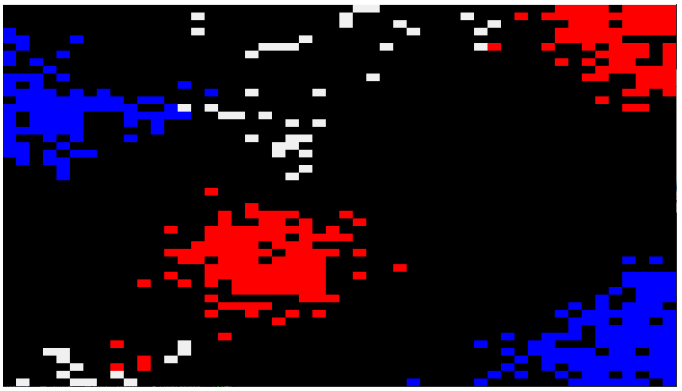

figure 13 : Tri au bout de 20 000 000 itérations

En implémentant une telle solution, on remarque que le tri est mieux réalisé en figure 13 au bout de 20 000 000 d'itérations.

Enfin, nous avons testé notre modélisation avec un système comprenant plus d'agent (40 à la place de 20) et d'objet C (200 à la place de 50).

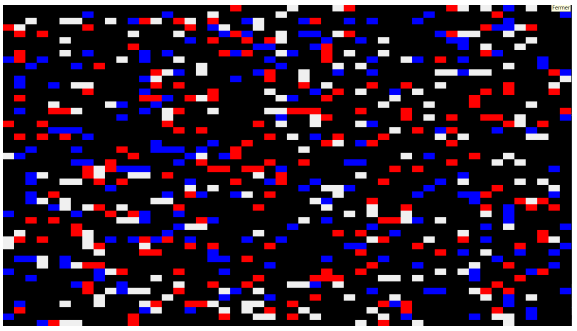

figure 14 : Tri au bout de 0 itération

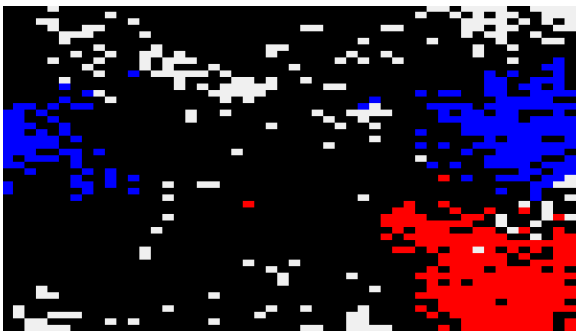

figure 14 : Tri au bout de 35 000 000 itérations

Sur les figures 14 et 15, on remarque que le tri semble se passer correctement néanmoins, il est beaucoup plus lent. Mais sur la figure 14, on commence à observer des clusters.

Conclusion

Ainsi, nous venons de réaliser la simulation d'un système multi-agents qui tri collectivement des objets. Ce tri est soit réalisé par un agent seul lorsqu'il s'agit de déplacer un objet A ou B soit réaliser à deux agents lorsqu'il faut déplacer un objet C. Afin de réaliser un tel système, nous nous sommes inspirés du modèle de dépôt de phéromone observable dans la nature.

On observe que le tri des objets est plutôt correct, mais que l'ajout d'objet C le ralentit. En effet, la collaboration de deux robots est plus lente et nécessite donc plus d'itérations.