



Lyon 1

Sujet TP 2 SMA

Élève :
Abdellaoui Anas

Enseignants :
HASSAS Salima

Problématique

On modélise notre environnement par une grille où sont disposés aléatoirement N_A objet de type A, N_B objet de type B, N_C objet de type C et N_{Agent} agents, avec une condition : sur la même case on ne peut pas avoir ni deux agents ni deux objets. Quand un robot tombe sur un objet C dans son environnement, il émet un signal (appel à l'aide) qui se code par une information propagée dans son voisinage (8 cases autour) sur une distance de diffusion du signal d_s (en nombre de cases : ex. les 8 cases autour de l'agent sur 1 distance de $d_s=2$: pour un agent se trouvant à l'emplacement (i,j) on aura toutes les cases se trouvant à $i+k, j+l$, avec $0 \leq k \leq 2$, et $0 \leq l \leq 2$). Les agents se déplacent aléatoirement sur les 8 directions disponibles et peuvent ramasser ou déposer un objet avec une probabilité dépendant de la proportion d'objets vus dans leur mémoire. $P_{prise} = (k_+ / (k_+ + f))^2$ et $P_{dépôt} = (f / (k_- + f))^2$ avec : k_+ et k_- des constantes et f représentant la proportion d'objet de même type A ou B dans l'environnement immédiat (voisinage de l'agent). Cette version diffère de la version 1 par l'ajout d'une troisième catégorie d'objet, les objets C. Ceux-ci sont trop lourds et un agent a besoin de l'aide d'un autre agent pour le déplacer, d'où l'introduction de l'aspect collaboration et communication entre les agents.

Implémentation

Dans l'implémentation je garde la même structure de la première version, avec l'ajout de quelques fonctionnalités sur les classes :

- Cell : l'ajout d'un nouvel attribut `taux`` représentant le taux de phéromone de chaque cellule. Cet attribut requiert des nouvelles méthodes, une méthode de dissipation de la phéromone avec un taux `r`` et les setters/getters habituels.

- Agent : Afin de modéliser l'appel à l'aide d'un agent, nous avons décidé d'utiliser le système de phéromones. On modélise le comportement d'agent par un état, à partir de cette dernière il peut réagir.

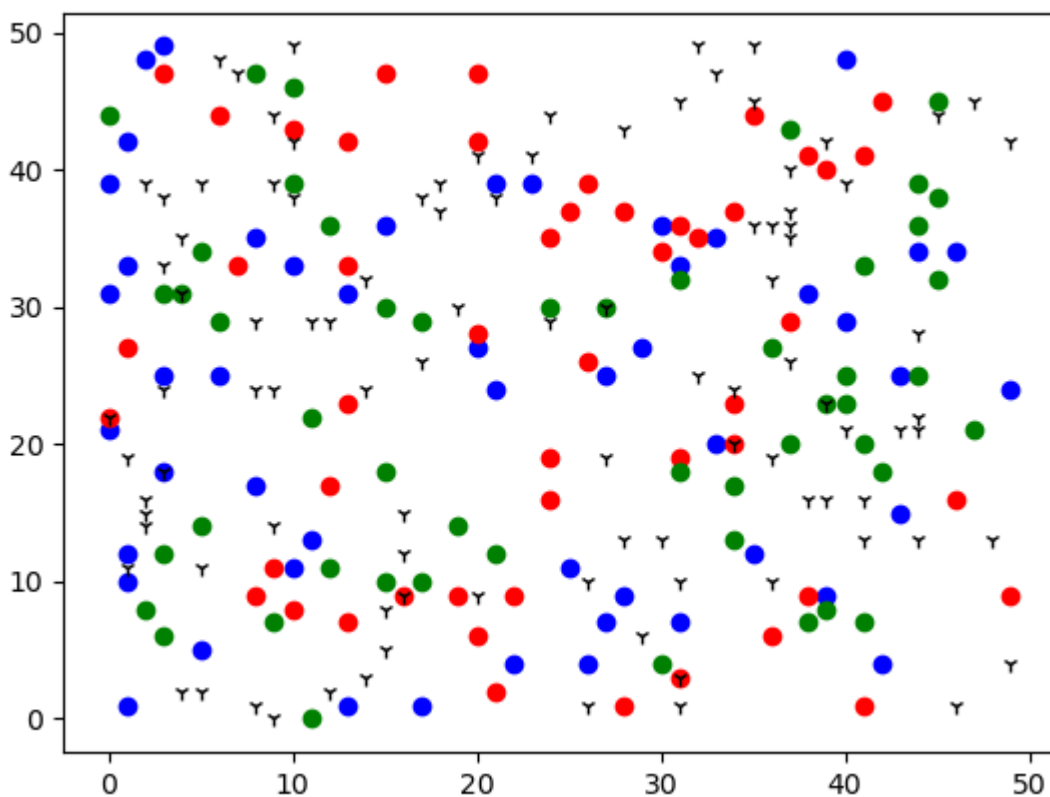
- **help** : L'agent dans cet état reste immobile sur un objet C et diffuse de la phéromone sur les cases voisines.
- **follow** : L'agent suit son leader et attend que celui-ci pose son objet.
- **Looking for** : L'agent cherche l'émetteur de phéromone en se déplaçant vers les cellules possédant un taux de phéromone plus élevé. Si il se trouve déjà sur le maximum alors il bascule en `free`` (cela veut dire qu'un autre agent a déjà aidé celui qui émettait), si il trouve celui qui émettait alors il bascule en `follow`` et l'émetteur bascule en `leader``.
- **free** : Les agents fonctionnent comme dans la version 1 (mouvement aléatoire et probabilité de prise/dépôt). Cependant s'il tombe sur un objet C et qu'il souhaite le ramasser il bascule dans l'état `help``. De plus s'il sent de la phéromone à un taux assez élevé il bascule dans l'état `Looking for``.
- **leader** : L'agent se comporte comme en `free`` mais il porte un objet C, un agent le suit (l'aide) et

s'il souhaite déposer un objet alors il le dépose et le suiveur ainsi que lui-même bascule en `free`. Dans cette deuxième version, les agents ont donc deux attributs supplémentaire, `leader` et `follower` qui représente respectivement le leader (si l'agent est un follower) et le follower (si l'agent est un leader).

Résultats

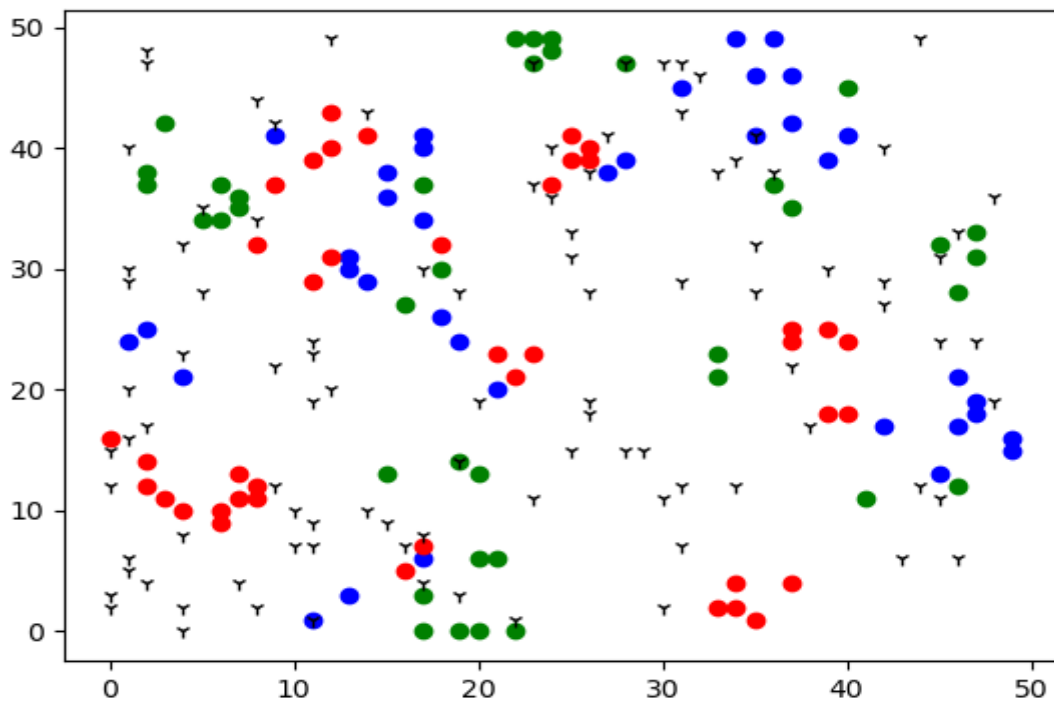
- Nombre de tours =0

Taille mémoire = 10



- Nombre de tours =1000

Taille mémoire = 10



- Nombre de tours =5000

Taille mémoire = 10

