Rapport de TP de SMA

Tri collectif

Sommaire

# Introduction

Dans le cadre du cours de SMA, nous avons implémenté l’algorithme de tri collectif décrit dans l’article « THE DYNAMICS OF COLLECTIVE SORTING ROBOT-LIKE ANTS AND ANT-LIKE ROBOTS ». Dans ce système, plusieurs agents sont placés sur une grille et sont chargés de trier des blocs selon leur type (A ou B). L’objectif final est d’obtenir deux types de colonies distincts : des colonies de blocs A et des colonies de blocs B. Ainsi, les agents sont censés éviter au maximum les mélanges de blocs au sein d’une même colonie.

Dans un premier temps, nous avons procédé à l’implémentation même du système multi-agents que nous avons réalisé en Java. Les détails concernant cette partie sont disponibles dans le README.md situé à la racine du projet.

Une fois le système implémenté, nous avons réalisé diverses statistiques afin d’évaluer l’influence des paramètres d’entrée sur les résultats obtenus après simulation. Un exemple de ces statistiques se trouve dans le dossier *sample/* du projet ; c’est ce fichier que nous prendrons pour référence dans la suite de ce rapport.

Nous analyserons successivement l’influence des différents paramètres sur les résultats obtenus finaux obtenus en nous basant principalement sur 3 critères :

* La proportion de blocs A voisins de blocs B,
* Le nombre de colonies formées par les agents,
* La proportion de blocs posés dans l’environnement.

# Valeurs de référence

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nombre de blocs A** | **Nombre de blocs B** | **Nombre de lignes de la grille** | **Nombre de colonnes de la grille** | **Nombre d’agents** | **Taille de la mémoire des agents** | **Nombre de mouvements successifs des agents** | **K +** | **K -** | **Erreur** |
| 100 | 100 | 40 | 40 | 20 | 15 | 1 | 0,1 | 0,3 | 0 |

En prenant comme paramètres par défaut les valeurs définies dans le tableau ci-dessus, nous avons les résultats suivants :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Nombre de blocs A posés** | **Nombre de blocs B posés** | **Proportion de blocs A voisins de blocs A** | **Proportion de blocs A voisins de blocs B** | **Proportion de blocs B voisins de blocs B** | **Nombre de colonies** | **Taille d’une colonie** | **Proportion de A par colonie** | **Proportion de B par colonie** |
| **Moyenne sur 10 exécutions** | 98 | 99,2 | 0,5317 | 0,0006 | 0,4677 | 27 | 7,42 | 0,44 | 0,56 |

# Nombre de blocs

Faisons tout d’abord varier le nombre de blocs présents dans l’environnement. Nous garderons dans un premier temps 50% de blocs A et 50% de blocs B, ce qui nous permet d’obtenir les graphiques suivants :

Ainsi, nous remarquons avec le graphe de gauche que plus il y a de blocs sur la grille, plus le nombre de colonies augmente, et ce de manière linéaire pour les valeurs testées (20, 200, 400 et 600 blocs). Le graphique de droite nous permet quant à lui d’observer que la taille des colonies augmente elle aussi linéairement avec le nombre de blocs.

Par ailleurs, ajouter des blocs à la grille augmente la proportion d’erreurs de tri (c’est-à-dire le nombre de blocs A voisins de blocs B). Ceci est une conséquence logique : en effet, plus il y a de blocs sur la grille, plus un bloc A a de chances de se retrouver à côté d’un autre bloc, potentiellement de type B.

Faisons maintenant varier les proportions de blocs A par rapport aux blocs B. Pour cela, nous considérerons 200 blocs de type B puis successivement 10, 100 et 300 blocs de type A.

Nous constatons ainsi que réduire le nombre de blocs A permet d’avoir moins d’erreurs de voisinage, ce qui est logique puisque comme il y a moins de blocs A, ces derniers ont une probabilité moindre de se retrouver à côté de blocs B. De même, moins il y a de blocs A, plus les colonies sont petites (moyenne de 3,6 blocs pour 10 blocs A et 300 blocs B contre 7,42 dans le cas de référence). Concernant le nombre de colonies, la différence de proportion entre blocs A et B ne semble pas donner de résultats exploitables.

Les mêmes observations peuvent être faites si l’on modifie le nombre de blocs B en gardant le nombre de blocs A constant.

# Taille de la grille

Intéressons-nous maintenant aux résultats obtenus lorsque nous faisons varier la taille de la grille sur laquelle évoluent les agents. En simulant l’exécution avec des grilles de tailles 20x20 (blocs recouvrant 50% de la grille), 40x40 (blocs recouvrant 12,5% de la grille) et 60x60 (blocs recouvrant 5,56% de la grille), nous avons généré les graphes suivants :

Nous remarquons que le nombre de colonies augmente avec la taille de la grille, ce qui est logique puisque plus l’environnement est vaste, plus les blocs sont libres d’être éparpillés et éloignés les uns des autres. De la même manière, comme il y a plus de colonies, ces dernières sont de plus petite taille comme nous le constatons sur le graphe de droite. L’évolution du nombre et de la taille des colonies semble pouvoir être approchée par une fonction logarithmique, comme le montrent les courbes de tendance ajoutées aux graphiques.

Si nous étudions les erreurs de voisinage commises en fonction de la taille de la grille, nous observons qu’elles sont plus nombreuses pour une grille de petite taille mais qu’elles se stabilisent lorsque l’environnement d’agrandit. Il faut donc que la grille soit suffisamment grande par rapport au nombre de blocs pour permettre de minimiser les erreurs de voisinage.

Si nous faisons maintenant varier le nombre de lignes indépendamment du nombre de colonnes (avec successivement 40, 50 et 100 lignes pour 40 colonnes), nous pouvons tirer les mêmes conclusions concernant les colonies, comme le démontrent les graphes ci-contre.

De plus, les erreurs de voisinage sont inférieures à 0,3% pour les valeurs testées, ce qui confirme l’observation faite précédemment selon laquelle en-deça d’un certain pourcentage de blocs sur la grille l’erreur commise est négligeable.

# Nombre d’agents

# Mémoire des agents

# Nombre de mouvements successifs des agents

# K +

# K –

# Erreur