Rapport de TP de SMA

Tri collectif

Présenté par Dylan MARSANGY et Laura PHILIBERT

MARSANGY Dylan – 11516569
PHILIBERT Laura - 11503975

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. Des exemples de ces statistiques se trouvent dans le dossier *doc/rendu/sample/* du projet ; ce sont ces 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 :

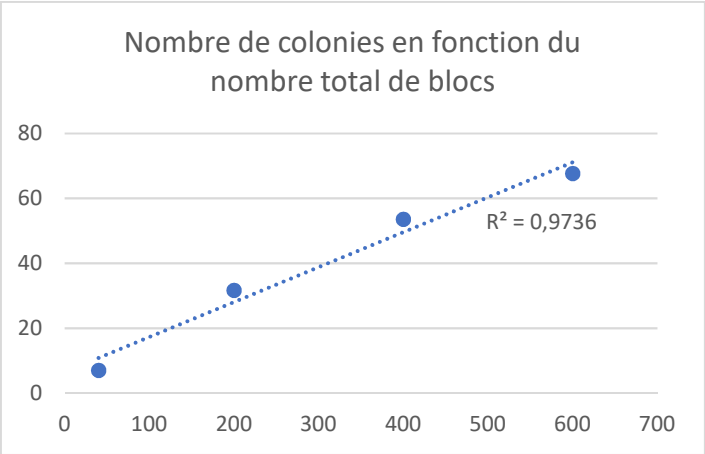
	Nombre de blocs A posés	Nombre de blocs B posés	Proportion de blocs A voisins de blocs A	Proportion de blocs A voisins de blocs B	Proportion de blocs B voisins de blocs B	Nombre de colonies	Taille d'une colonie	Proportion de A par colonie	Proportion de B par colonie
Moyenne sur 10 exécutions	98	99,2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

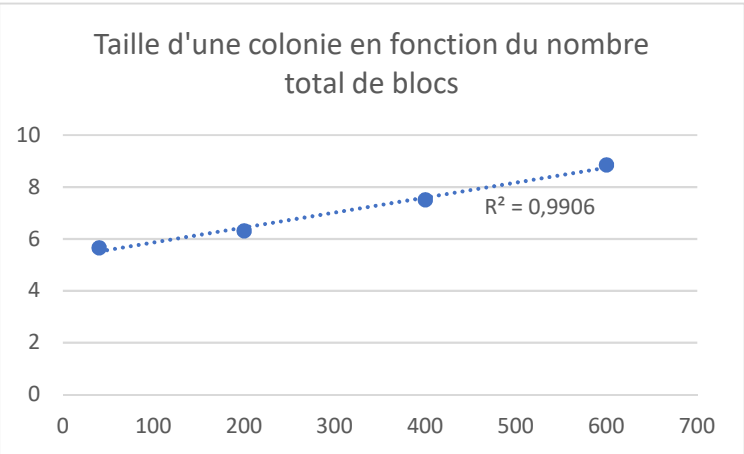
Tous les algorithmes ont été exécutés avec le même nombre d'itérations, c'est-à-dire 1 600 000 itérations.

Nombre de blocs

1. Observations

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 :

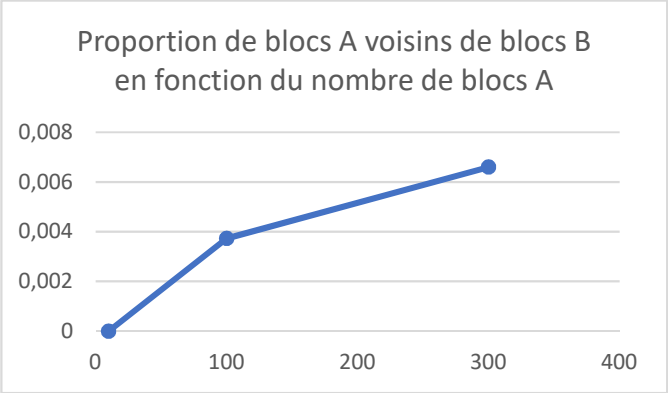


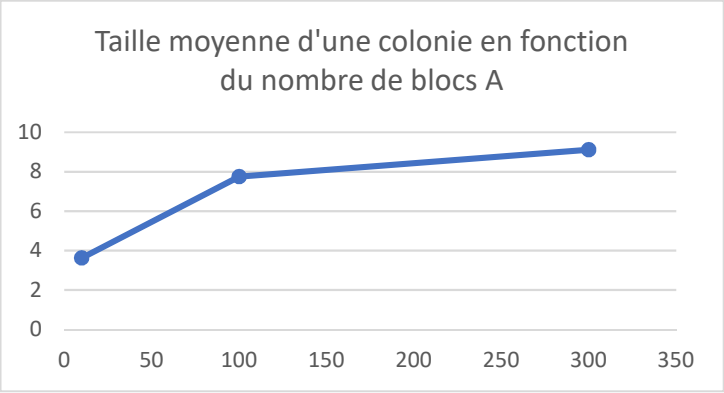


Ainsi, nous remarquons avec le graphe de gauche que plus il y a de blocs sur la grille, plus le nombre de colonies augmente, et ce de manière linéaire pour les valeurs testées (20, 200, 400 et 600 blocs). Le graphique de droite nous permet quant à lui d'observer que la taille des colonies augmente elle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

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.

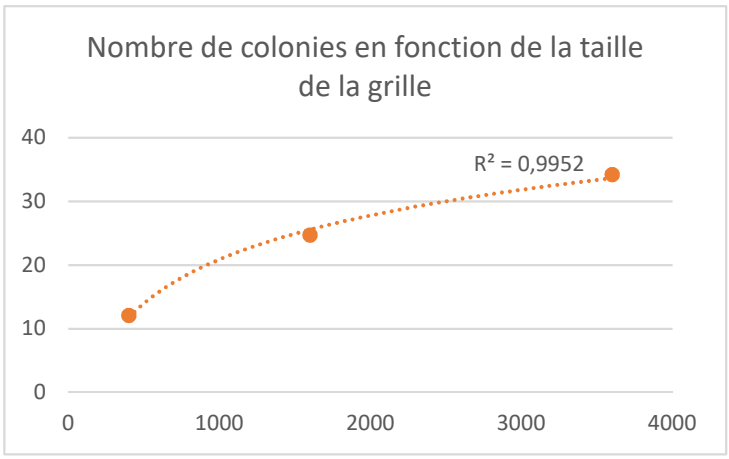
2. Analyse

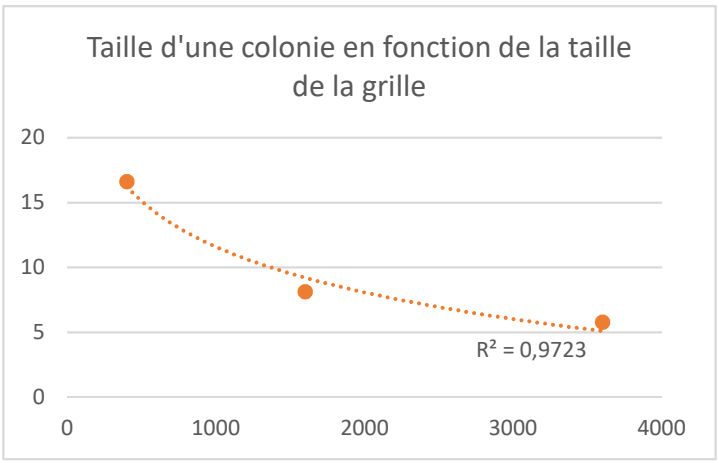
Finalement, plus il y a de blocs dans l'environnement, plus la taille des colonies tend à augmenter (puisqu'il y a plus de blocs, les colonies sont fatalement plus étendues). Si nous gardons les mêmes proportions entre blocs de type A et blocs de type B, nous constatons également que le nombre de colonies formées par les agents augmente sans que les erreurs de voisinage ne soient impactées. En revanche, si nous introduisons des écarts entre les proportions de blocs de chaque type, la proportion d'erreur de voisinage augmentera mais le nombre de colonies ne changera pas significativement.

Taille de la grille

1. Observations

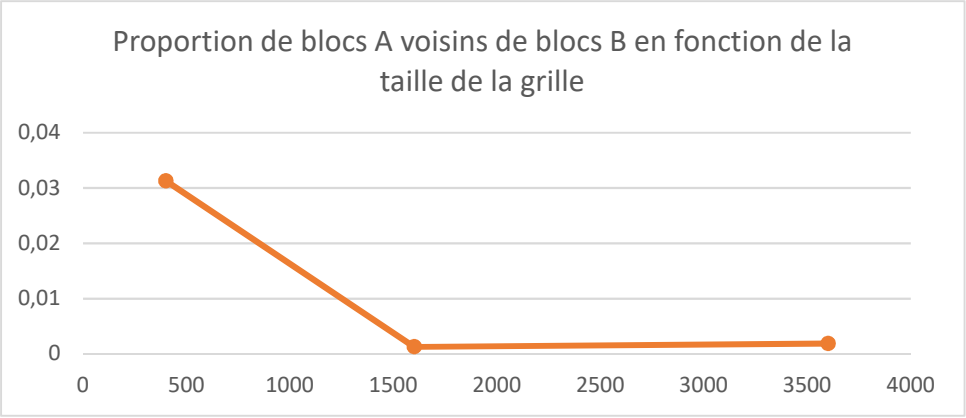
Intéressons-nous maintenant aux résultats obtenus lorsque nous faisons varier la taille de la grille sur laquelle évoluent les agents. En simulant l'exécution avec des grilles de tailles 20x20 (blocs recouvrant 50% de la grille), 40x40 (blocs recouvrant 12,5% de la grille) et 60x60 (blocs recouvrant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



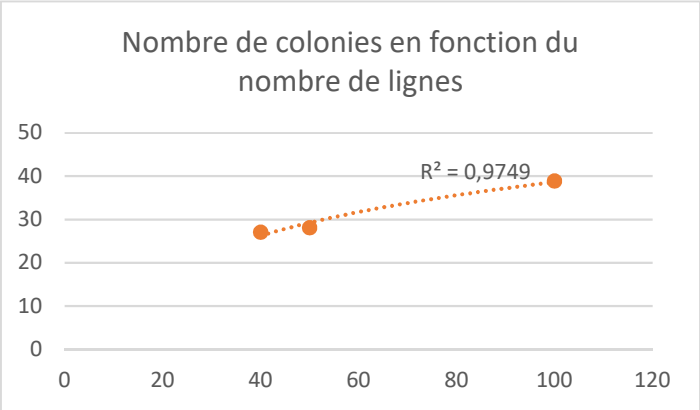


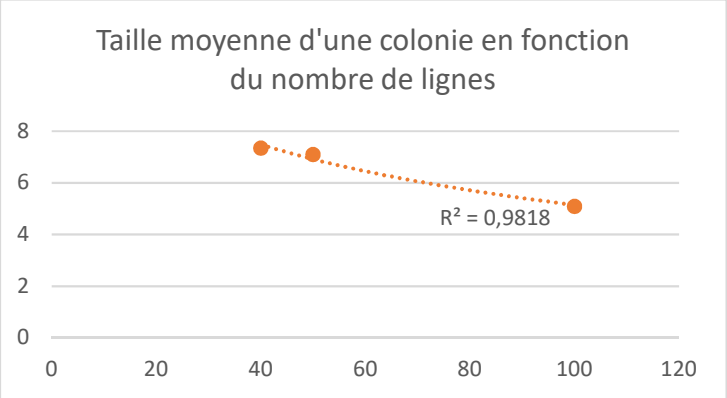
Nous remarquons que le nombre de colonies augmente avec la taille de la grille, ce qui est logique puisque plus l'environnement est vaste, plus les blocs sont libres d'être éparpillés et éloignés les uns des autres. De la même manière, comme il y a plus de colonies, ces dernières sont de plus petite 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 s'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-deçà d'un certain pourcentage de blocs sur la grille l'erreur commise est négligeable.

Les mêmes conclusions peuvent être faites si nous faisons varier le nombre de colonnes indépendamment du nombre de lignes de la grille.

2. Analyse

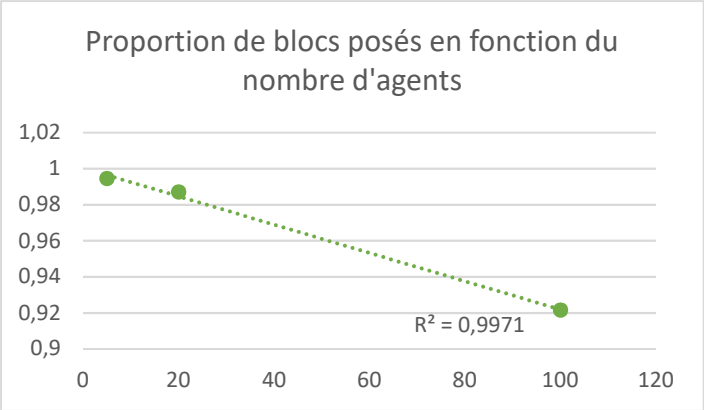
La taille de la grille utilisée pour représenter l'environnement impacte donc les colonies formées par les agents. En effet, plus la grille sera de petite taille, moins il y aura de colonies et plus elles comporteront de blocs. Par ailleurs, si nous réduisons trop les dimensions de la grille par rapport au nombre de blocs, les erreurs de voisinage augmenteront significativement, conséquence du manque de place dans l'environnement.

Nombre d'agents

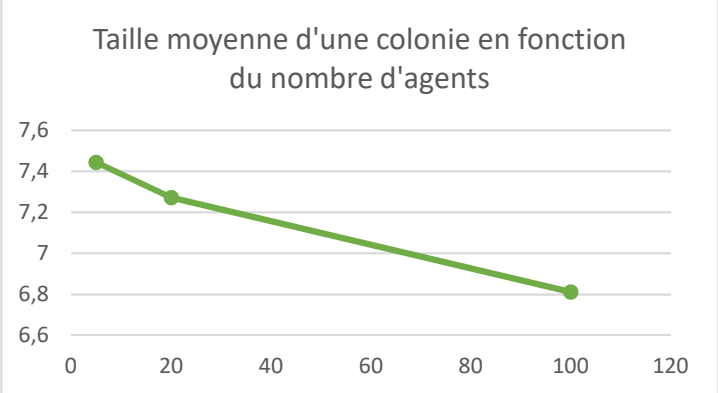
1. Observations

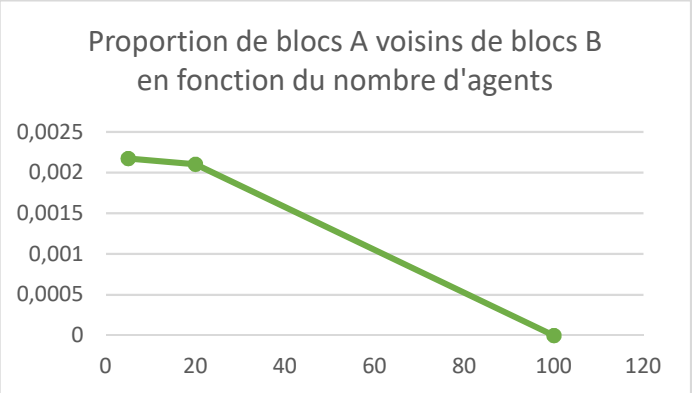
Étudions maintenant l'influence du nombre d'agents utilisés dans l'environnement sur les résultats obtenus. Nous considérerons successivement l'emploi de 5, 20 puis 100 agents.

Tout d'abord, nous notons que plus le nombre d'agents est important, moins il y a de blocs posés en fin de simulation. Ce résultat s'explique par le fait que plus il y a d'agents, plus les blocs sont susceptibles d'être soulevés à un instant t. L'évolution de la proportion de blocs posés dans l'environnement en fonction de la quantité d'agents peut être approximée par une fonction linéaire comme le montre la courbe de tendance ajoutée au graphe.



Nous pouvons par ailleurs observer que la taille des colonies tend à diminuer lorsque le nombre d'agents augmente, tout comme les erreurs de voisinage.





2. Analyse

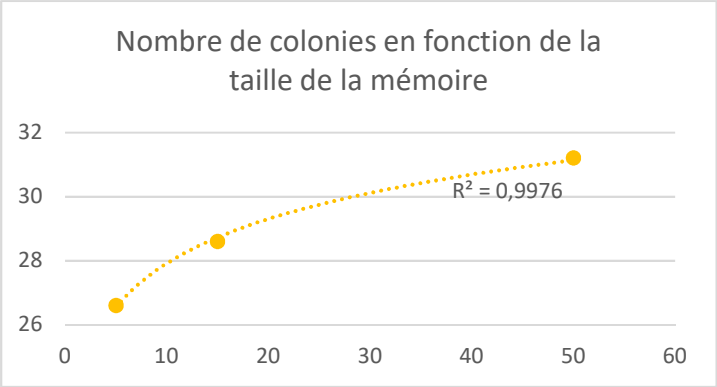
À la finalité, même si accroître le nombre d'agents permet à première vue de réduire les erreurs de voisinage, il convient de nuancer ce résultat par le fait que le nombre de blocs posés, et donc analysables en fin de simulation, décroît avec la quantité d'agents. Il est donc difficile d'affirmer que les agents sont plus efficaces lorsqu'ils sont plus nombreux simplement avec ces observations.

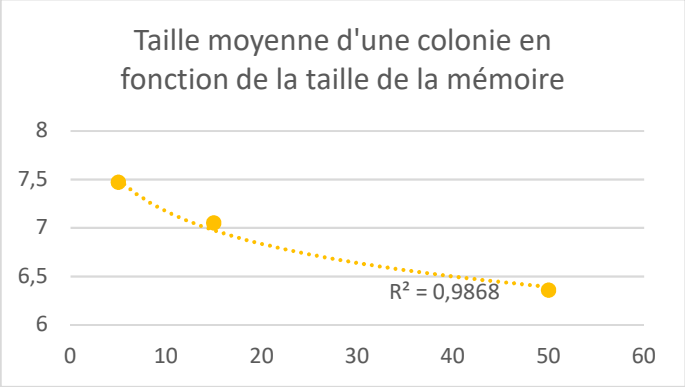
Mémoire des agents

1. Observations

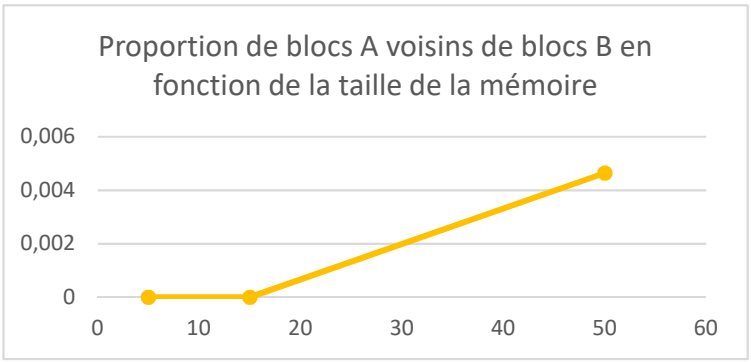
Faisons à présent varier la taille de la mémoire des agents. Nous prendrons pour nos tests des mémoires de taille 5, 15 et 50.

Dans un premier temps, nous remarquons que plus les agents ont une mémoire de taille importante, plus le nombre de colonies formées augmente. La conséquence directe de cette augmentation est une baisse du nombre moyen de blocs par colonies, comme nous l'observons sur les 2 graphes ci-contre. Les évolutions de ces 2 mesures en fonction de la taille de la mémoire peuvent être approximées par des fonctions logarithmiques.





Par ailleurs, nous avons pu constater qu'avec une mémoire trop importante, les erreurs de voisinage augmentaient. En effet, avec des tailles de mémoire de 5 ou 15 blocs, nous avons n'observons aucune erreur alors qu'avec une mémoire de 50 blocs, nous avons une erreur moyenne de 0,45%.



Ce résultat est logique puisque plus la taille de la mémoire est grande, plus l'agent se souvient d'anciens voisinages dans lesquels il ne se situe plus nécessairement, ce qui influencera négativement sa décision de prise.

2. Analyse

Nous pouvons tirer plusieurs conjectures des observations réalisées. Une mémoire de faible capacité permet de réduire à la fois les erreurs de voisinage et le nombre de colonies dans l'environnement. A l'inverse, des agents dotées d'une mémoire de grande capacité formeront davantage de colonies et produiront plus d'erreurs de voisinage.

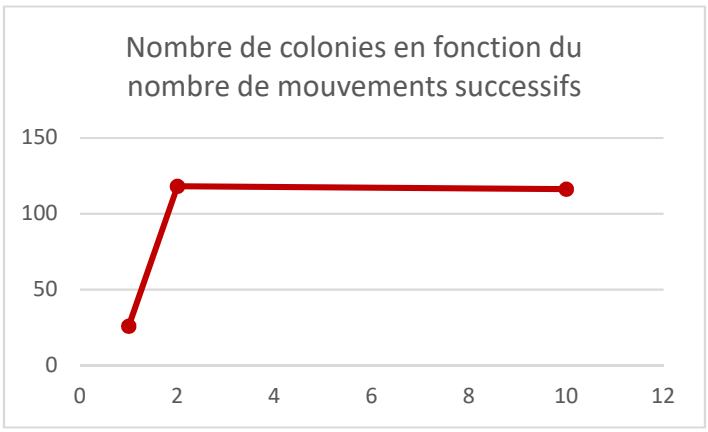
En conséquence, pour répondre à notre problématique, il vaut mieux utiliser une mémoire de **faible capacité** pour les agents.

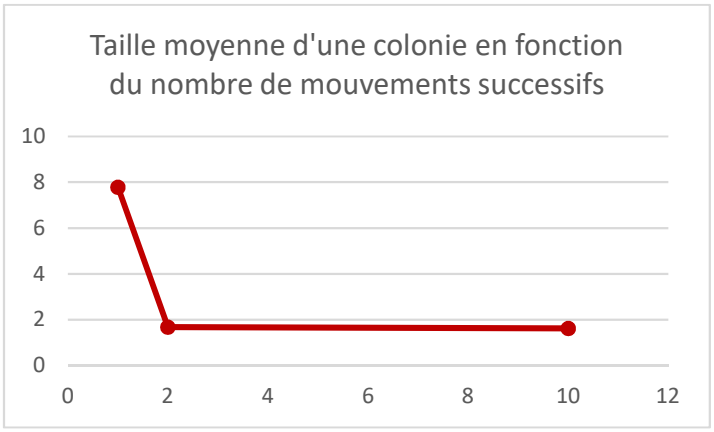
Nombre de mouvements successifs des agents

1. Observations

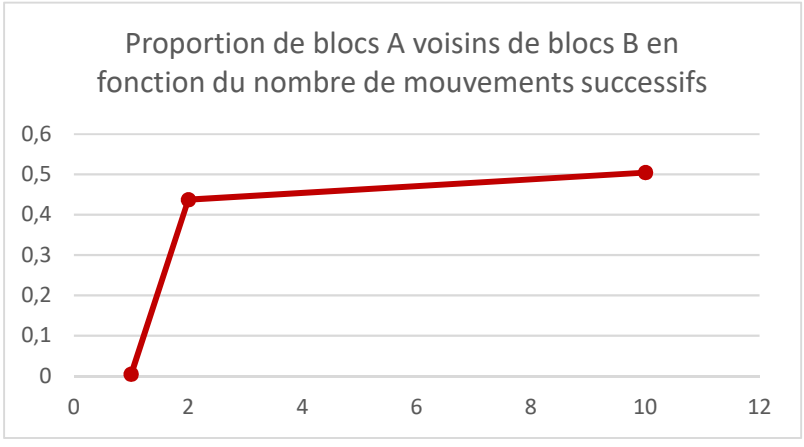
Nous allons à présent modifier le nombre de mouvements successifs des agents, autrement dit le nombre de cases qu'ils vont parcourir en une seule étape. Nous étudierons les résultats obtenus avec i = 1, i = 2 et i = 10.

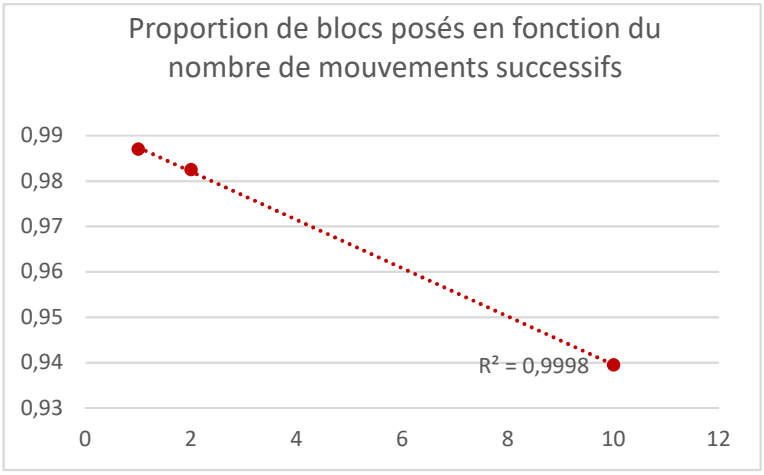
Au niveau du nombre et de la taille des colonies, nous notons une forte disparité entre les valeurs i = 1 et i = 2. En effet, dans le second cas, le nombre de colonies formées par les agents croît fortement, passant de 25,9 à 118,2. En conséquence, le nombre de blocs par colonie chute drastiquement de 7,8 à 1,7. Entre i = 2 et i = 10, les différences sont quasiment inexistantes, ce qui contraste avec les écarts observés entre des mouvements successifs de 1 ou de 2 cases.





Par ailleurs, la proportion d'erreurs de voisinage augmente elle aussi énormément lorsque nous passons de 1 à 2 mouvements successifs (0,45% d'erreur contre 43,7%). Pour i = 10, les erreurs de voisinage dépassent même les 50%. En parallèle, de moins en moins de blocs sont déposés par les agents au moment de la fin de la simulation. Là où nous avions 98,7% de blocs posés pour i = 1, nous n'en avons plus que 93,95% pour i = 10 (évolution linéaire). Ainsi, les erreurs de voisinage sont plus nombreuses alors même qu'il y a moins de blocs sur la grille.





2. Analyse

D'après les résultats que nous avons obtenus, nous pouvons affirmer qu'augmenter le nombre de mouvements successifs nuit aux résultats de bien des manières. Tout d'abord, les agents sont moins à même de déposer les blocs au moment de l'arrêt de la simulation mais ils créent également plus de colonies et introduisent davantage d'erreurs de voisinage.

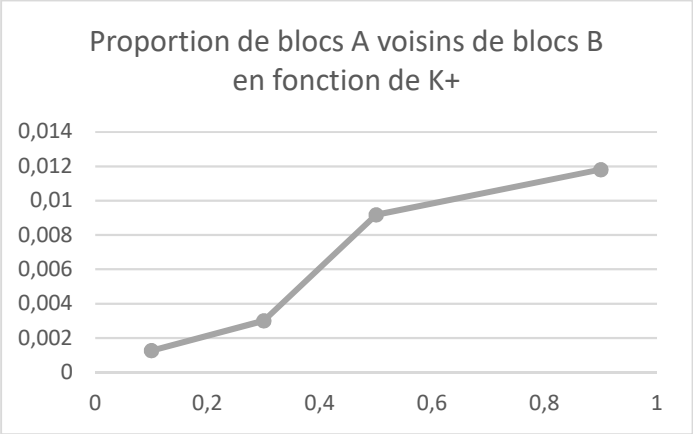
Il convient donc de conserver un nombre de mouvements successifs égal à **1** pour maximiser les chances de réussite.

K+

1. Observations

Intéressons-nous à présent aux différences de résultats obtenues lors de la modification du paramètre K+. Pour nos tests, nous avons choisi d'utiliser les valeurs K+ = 0.1, K+ = 0.3, K+ = 0.5 et K+ = 0.9.

Nous avons ainsi pu constater que le nombre et la taille des colonies formées par les agents ne changeaient pas de manière significative en fonction de ce paramètre. En revanche, nous avons remarqué que plus K+ était proche de 1, plus la proportion d'erreurs de voisinage augmentait, jusqu'à dépasser 1% pour K+ = 0.9.



En parallèle, nous observons que de moins en moins de blocs sont posés en fin de simulation lorsque K+ augmente alors que les erreurs continuent de croître.

Ce résultat est logique étant donnée la formule de prise des blocs. Plus K+ augmente, plus la valeur de f (basée sur la perception de l'agent) est négligeable. Ainsi, une valeur importante de K+ revient à ne pas prendre en compte le voisinage de l'agent, alors qu'il est à la base de la décision de prise pour une efficacité optimale.

2. Analyse

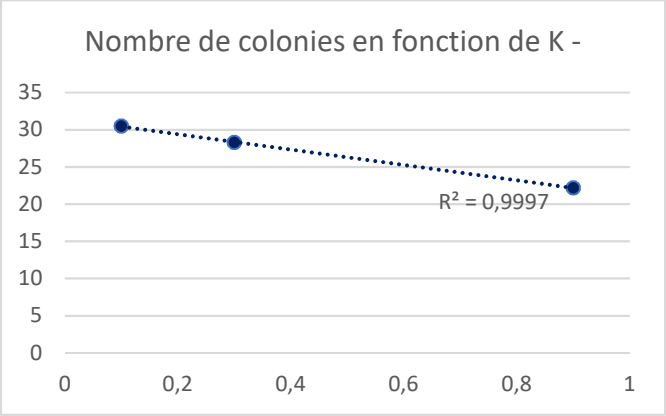
Finalement, nous pouvons conclure que choisir une valeur de K+ **proche de 0** est bénéfique dans notre système. En effet, cela permet de réduire les erreurs de voisinage sans impacter le nombre et la taille des colonies formées de manière significative. Bien évidemment, elle ne doit pas être égale à 0, auquel cas les agents ne prendraient jamais de blocs.

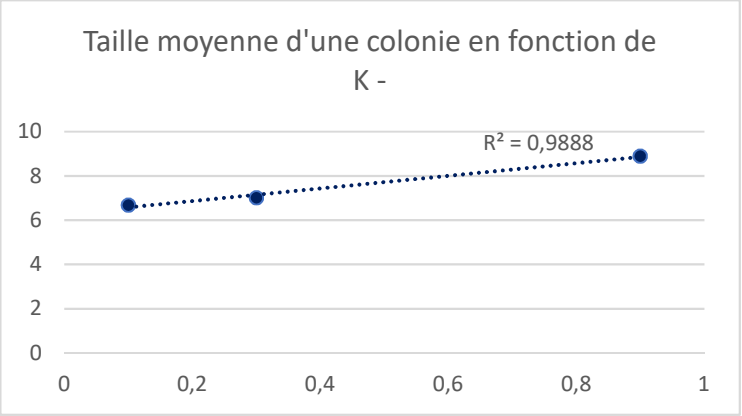
K–

1. Observations

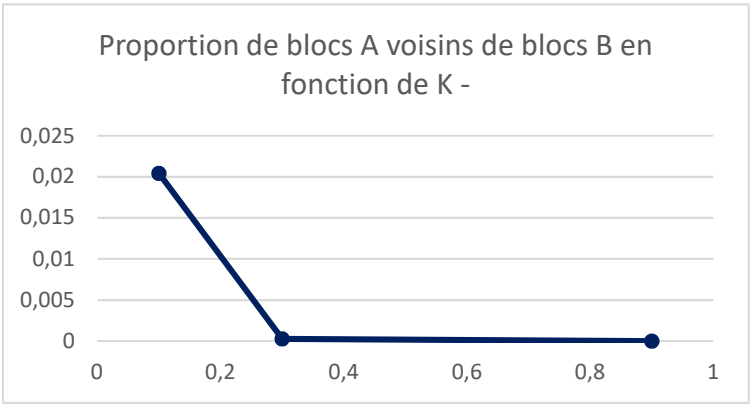
Nous allons dans cette partie nous intéresser aux valeurs possibles pour le paramètre K-. Nous avons pour cela relevé les résultats obtenus avec les valeurs K- = 0.1, K- = 0.3 et K- = 0.9.

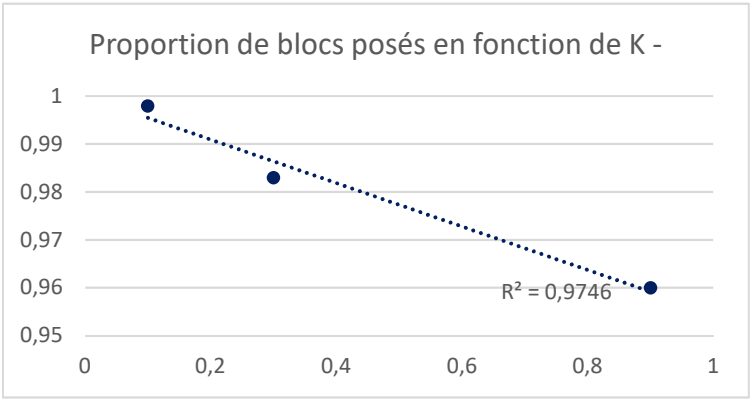
Dans un premier temps, nous nous apercevons que lorsque K- s'approche de 1, le nombre de colonies diminue tandis que leur taille augmente, et ce de manière linéaire (cf. courbes de tendance sur les graphiques ci-dessous).





De plus, lorsque K- croît, le pourcentage d'erreurs de voisinage chute, passant de 2% pour K- = 0.1 à 0% pour K- = 0.9. En parallèle, nous constatons que la proportion de blocs posés en fin de simulation est inversement proportionnelle à K-, allant de 99,8% pour K- = 0.1 à 96% pour K- = 0.9. Ainsi, même si les erreurs de voisinage sont, à priori, réduites, il est difficile de tirer de véritables conclusions puisque de plus en plus de blocs sont soulevés par les agents au moment de la fin de la simulation.





2. Analyse

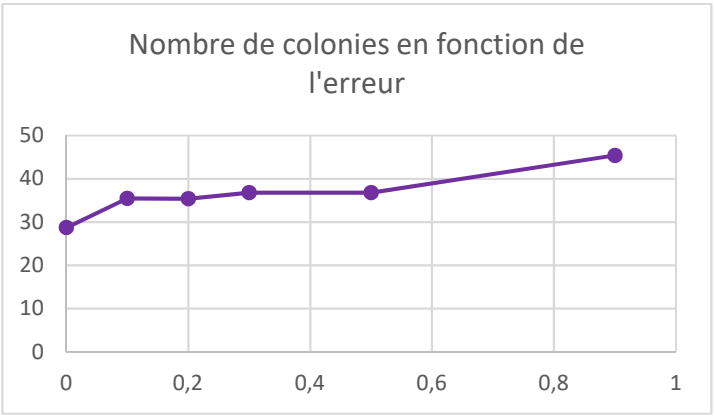
Pour conclure, nous remarquons qu'augmenter la valeur de K- peut, certes, se révéler efficace pour réduire à la fois les erreurs de voisinage et le nombre de colonies mais que ces améliorations se font au détriment du nombre de blocs posés par les agents. La solution la plus adaptée semble alors d'utiliser **K- = 0.3** puisque c'est cette valeur offre le meilleur compromis parmi celles que nous avons testées.

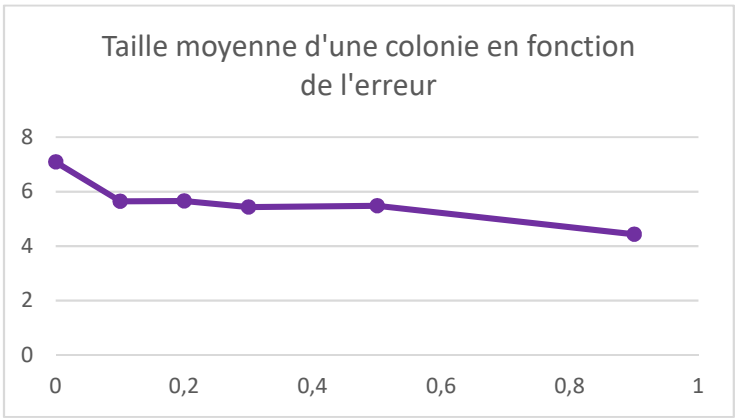
Erreur

1. Observations

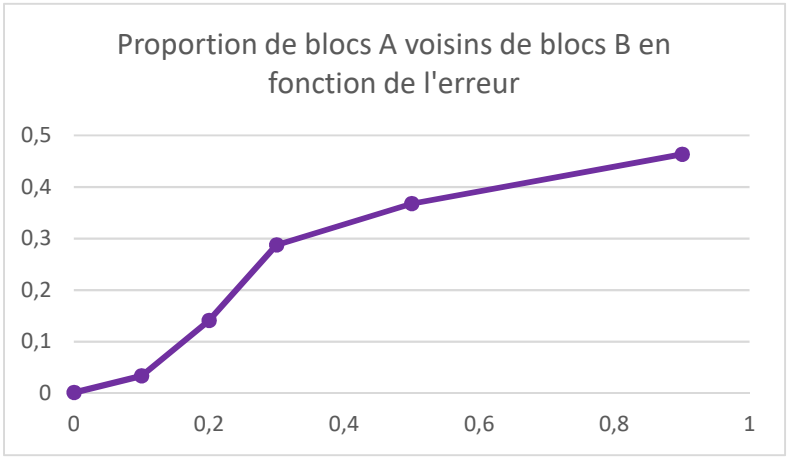
Nous allons maintenant ajouter des erreurs dans la mémoire des agents. A chaque prise de bloc, l'agent aura alors une certaine probabilité de se tromper et d'ajouter le mauvais bloc à sa mémoire. Pour nos tests, nous considérerons des probabilités de 0, 0.1, 0.2, 0.3, 0.5, et 0.9.

Pour commencer, nous voyons avec les graphiques ci-dessous qu'augmenter l'erreur influence à la fois le nombre et la taille des colonies formées. En effet, plus l'erreur est importante, plus les colonies sont nombreuses et moins densément peuplées. Ainsi, nous passons de 28,7 colonies pour une erreur de 0 à 45,4 colonies en moyenne pour une erreur de 0,9.





D'autre part, nous avons observé que plus l'erreur était proche de 1, plus la proportion d'erreurs de voisinage augmentait. Alors que nous avons un pourcentage de A voisins de B de 0,14% pour e = 0, nous passons à 46,3% pour e = 0.9.



2. Analyse

Ajouter la notion d'erreur ne s'est pas révélé bénéfique pour notre algorithme. En effet, nous avons considérablement augmenté le pourcentage d'erreurs de tri tout en créant plus de colonies. Il vaut donc mieux laisser l'erreur à **0** pour éviter une dégradation des résultats.

Conclusion

Pour conclure, l'analyse des différents paramètres nous a permis de tirer plusieurs conjectures quant aux valeurs à privilégier pour obtenir de bons résultats. Le tableau ci-dessous récapitule les observations faites paramètre par paramètre.

Nombre de blocs A	**Nombre de blocs B**	**Taille de la grille**	**Nombre d'agents**	**Taille de la mémoire des agents**	**Nombre de mouvements successifs des agents**	**K +**	**K -**	**Erreur**
100	Égal au nombre de blocs A	Telle qu'il y ait environ 12,5% de blocs	Inférieur à un dixième du nombre total de blocs	Petite (5 ou 15)	1	Proche de 0 (0,1)	0,3	0

D'un point de vue personnel, ce TP nous a permis d'approfondir nos connaissances en systèmes multi-agents. Nous avons également pu voir, grâce à l'analyse des résultats obtenus lors des simulations, l'influence que pouvaient avoir les changements, en apparence anodins, des paramètres en entrée.