

UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

RENDEZ-VOUS SYNCHRONE DANS LES GRILLES

MÉMOIRE
PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN SCIENCES ET TECHNOLOGIES DE L'INFORMATION

PAR
ANAS TIANE

OCTOBRE 2013

Ce mémoire a été évalué par un jury composé des personnes suivantes :

Dr. Andrzej Pelc Directeur de recherche

Dr. Mohand Said Allili Président du jury et lecteur

Dr. Jurek Czyzowicz Lecteur

Mémoire accepté le : 5 Décembre 2013

*À mes parents qui ont tout sacrifié pour que je puisse me rendre là où je suis
aujourd'hui ...*

*À ma famille qui m'a toujours supportée, surtout durant les phases difficiles de cette
maîtrise*

*Au défunt, le professeur AbdelSalam Yassine, qui m'a incité et encourager à finaliser
cette maîtrise*

Remerciement

Je tiens fortement à remercier mon directeur de recherche Monsieur Andrzej Pelc pour son leadership et encadrement exceptionnels qui m'ont permis de me dépasser, acquérir de nouvelles connaissances et livrer un mémoire avec la mention excellent.

Table des matières

Liste des figures	iv
Liste des tableaux	v
Résumé	vii
1 Introduction	1
1.1 Le modèle	3
1.2 Les résultats	3
2 Revue de la littérature	5
2.1 Les algorithmes de rendez-vous déterministe	5
2.1.1 Rendez-vous déterministe synchrone dans un graphe fini	5
2.1.2 Rendez-vous déterministe asynchrone dans un graphe infini	6
2.2 Les algorithmes de rendez-vous aléatoires	8
2.3 Les algorithmes de rendez-vous utilisant des jetons	9
3 Les algorithmes	11
3.1 Les algorithmes aléatoires	11
3.1.1 La marche aléatoire avec retour jusqu'au rendez-vous	11
3.1.2 La marche aléatoire sans retour jusqu'au rendez-vous	11
3.2 Les algorithmes déterministes	12
3.2.1 Boîte à outils	12
3.2.2 Algorithme : Rencontre sur le bord de la grille	14
3.2.3 Algorithme : Rencontre au centre de la grille : Classe C	14
3.2.4 Algorithme : Rencontre au centre de la grille : Classe B	20

4	La simulation	24
4.1	Résultats pour la simulation du rendez-vous au bord d'une grille de taille $m \times k$, avec $2 < m \leq k$	24
4.1.1	Données d'entrée	24
4.1.2	Les résultats	26
4.2	Résultats pour la simulation du rendez-vous au centre d'une grille de taille $m \times k$, avec $2 < m \leq k$	31
4.2.1	Données d'entrée	31
4.2.2	Les résultats	31
4.3	Résultats pour la simulation du rendez-vous au centre d'une grille de taille $2 \times k$	36
4.3.1	Données d'entrée	36
4.3.2	Les résultats	37
4.4	Résultats pour la simulation du rendez-vous aléatoire avec et sans retour	39
4.4.1	Données d'entrée	39
4.4.2	Les résultats	40
4.4.3	Résultats pour la simulation du rendez-vous aléatoire avec retour	40
4.4.4	Résultats pour la simulation du rendez-vous aléatoire sans retour	44
5	Conclusion	49

Liste des figures

3.1	La procédure d'exploration à partir du coin de la grille à l'étape $s = 3$.	16
3.2	Une grille de taille 7×9 après avoir exécuté la procédure <code>exploration</code> à partir du coin de la grille et la procédure <code>compléter le quart</code> pour les quatres coins de la grille. Les nœuds marqués sur la carte partielle sont noirs.	19
3.3	Échelle de taille 2×7 .	20
4.1	Les cinq configurations de départ pour le rendez-vous au bord de la grille	25
4.2	Les trois configurations de départ pour le rendez-vous au centre de l'échelle	36

Liste des tableaux

2.1	Temps du rendez-vous en fonction du nombre de nœuds, de l'orientation de l'anneau et de la distance initiale qui sépare les agents	10
2.2	Temps du rendez-vous en fonction du moment de perte des jetons	10
3.1	la liste des caractéristiques à la fin de l'étape $e = 2$	22
3.2	les caractéristiques de centre de l'échelle	22
4.1	Résultats pour le rendez-vous au bord d'une grille de taille 5×10	27
4.2	Résultats pour le rendez-vous au bord d'une grille de taille 5×20	27
4.3	Résultats pour le rendez-vous au bord d'une grille de taille 5×50	28
4.4	Résultats pour le rendez-vous au bord d'une grille de taille 5×100	28
4.5	Résultats pour le rendez-vous au bord d'une grille de taille 10×10	29
4.6	Résultats pour le rendez-vous au bord d'une grille de taille 20×20	29
4.7	Résultats pour le rendez-vous au bord d'une grille de taille 50×50	30
4.8	Résultats pour le rendez-vous au bord d'une grille de taille 100×100	30
4.9	Résultats pour le rendez-vous au centre d'une grille de taille 5×11	32
4.10	Résultats pour le rendez-vous au centre d'une grille de taille 5×21	33
4.11	Résultats pour le rendez-vous au centre d'une grille de taille 5×51	33
4.12	Résultats pour le rendez-vous au centre d'une grille de taille 5×101	34
4.13	Résultats pour le rendez-vous au centre d'une grille de taille 11×11	34
4.14	Résultats pour le rendez-vous au centre d'une grille de taille 21×21	35
4.15	Résultats pour le rendez-vous au centre d'une grille de taille 51×51	35
4.16	Résultats pour le rendez-vous au centre de l'échelle de taille 2×5	38
4.17	Résultats pour le rendez-vous au centre de l'échelle de taille 2×11	38
4.18	Résultats pour le rendez-vous au centre de l'échelle de taille 2×15	39

4.19 Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 5 × 5	40
4.20 Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 5 × 10	41
4.21 Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 5 × 20	41
4.22 Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 5 × 50	42
4.23 Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 5 × 100	42
4.24 Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 10 × 10	43
4.25 Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 20 × 20	43
4.26 Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 50 × 50	44
4.27 Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 5 × 5	44
4.28 Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 5 × 10	45
4.29 Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 5 × 20	45
4.30 Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 5 × 50	46
4.31 Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 5 × 100	46
4.32 Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 10 × 10	47
4.33 Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 20 × 20	47
4.34 Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 50 × 50	48

Résumé

Deux agents mobiles qui commencent à deux nœuds arbitraires dans une grille de taille $m \times k$, avec $1 < m \leq k$, doivent réaliser un rendez-vous en se retrouvant en même temps dans le même nœud. Les agents ont des étiquettes différentes et se déplacent de façon synchrone : dans chaque ronde un agent peut rester dans le même nœud ou aller à un nœud adjacent. Le temps du rendez-vous c'est le nombre de rondes entre le début du deuxième agent et le rendez-vous. Les nœuds de la grille sont anonymes et les ports locaux de chaque nœud de la grille sont libellés de façon arbitraire avec des numéros distincts $\{0, \dots, d - 1\}$. d est appelé le degré du nœud v avec $d = 2, 3, 4$. La numérotation des ports est locale, cela veut dire qu'il n'existe aucune relation entre les numéros de ports de deux nœuds distincts de la grille. Lorsqu'un nœud est visité, l'agent reconnaît son degré. L'agent mémorise le numéro de port par lequel il est entré dans le nœud et peut choisir le numéro de port par lequel il quittera le nœud. On considère deux catégories d'algorithmes, la première aléatoire et la deuxième déterministe. Pour les algorithmes aléatoires, on présente deux variantes, à savoir le rendez-vous aléatoire avec retour et sans retour. Pour les algorithmes déterministes, on présente deux variantes pour les grilles de taille $m \times k$ pour $1 < m \leq k$, à savoir le rendez-vous déterministe sur le bord de la grille et le rendez-vous déterministe au centre de la grille. Nous présentons deux algorithmes déterministes de rendez-vous pour les grilles $m \times k$, où $2 < m \leq k$: un qui fonctionne en temps $O(2^m + m^2k)$ et l'autre qui fonctionne en temps $O((2^m + k^2)l)$, où l est la plus petite des étiquettes. Ce premier fonctionne pour les grilles des dimensions impaires. Pour les grilles de taille $2 \times k$ on présente un algorithme déterministe qui se réalise en temps de $O(2^k l)$ où l est la plus petite des étiquettes.

Mot clés : : algorithmes, rendez-vous, agent mobile, grille.

Chapitre 1

Introduction

La rencontre de deux ou plusieurs agents mobiles dans un réseau dont la topologie est modélisée par un graphe fut pendant les dernières décennies le sujet d'intérêt de plusieurs recherches. L'article [14] donne des exemples d'utilisation des agents mobiles qui mettent en relief leurs forces et avantages surtout au niveau des applications qui opèrent dans un réseau complexe. Ainsi les agents mobiles peuvent être utilisés dans le domaine de :

- Maintenance des réseaux : dans un réseau hétérogène et complexe un agent mobile peut facilement visiter les nœuds du réseau pour vérifier leurs disponibilités et mettre à jour leurs logiciels.
- Orchestration des transactions simultanées et distribuées : dans le domaine du commerce électronique, une transaction peut être composée de plusieurs processus qui roulent de façon simultanée, par exemple : la vérification de l'adresse de l'utilisateur, la vérification du numéro de la carte de crédit, la vérification de la disponibilité des chambres, etc... Un agent mobile peut jouer le rôle de l'orchestrateur en naviguant entre les applications et synchroniser les résultats des transactions.

À travers les exemples mentionnés ci-haut, les agents mobiles peuvent être décrits comme étant des entités :

- logicielles autonomes de leur programme créateur, capables de prendre des décisions sans avoir recours à une entité référence centrale.
- mobiles, capables de naviguer d'un nœud à un autre dans un domaine distribué et complexe.
- interactives, qui interagissent avec les nœuds ou programmes visités, rapportent ou mettent à jour leur états.

- intelligentes, capables de s'adapter aux nouvelles situations et surtout capables de bâtir une connaissance à partir de l'historique de leurs parcours.

Les agents mobiles comptent plusieurs avantages, entre autre leur efficacité au niveau de leur format compact (logiciel plus mémoire). Ils participent à l'amélioration de la performance au niveau de la vitesse d'exécution et l'utilisation de la bande passante du réseau. Un agent mobile visite n nœuds en n étapes alors qu'un programme stationnaire central le fera en $2 \times n$ étapes. Les agents mobiles offrent des possibilités de recouvrement et donne une latitude au niveau de la tolérance à l'erreur. Un agent peut se déplacer et remplacer un nœud défectueux et permettre ainsi le maintien et la disponibilité continue du service. Les agents les plus avancées sont capables de bâtir leur propre base de connaissance en se basant sur l'historique de leur parcours et de monter leur propre processus de prise de décision.

Ceci dit les agents mobiles peuvent démontrer quelques faiblesses, au niveau de la sécurité sachant que c'est l'un des points fort démontrés par les recherches qui prônent l'utilisation des agents fixes. L'article [21] donne des exemples d'applications qui utilisent des agents mobiles sans compromettre la sécurité. Les articles [3, 4, 5] démontrent l'avantage d'utiliser des agents mobiles versus des agents fixes surtout au niveau de la tolérance à l'erreur.

Les recherches qui portent sur le rendez-vous des agents mobiles tentent de minimiser le temps de rencontre en prenant en considération un ou plusieurs des facteurs suivants : la topologie de l'espace de navigation (fini, infini, arbre, graphe, plan, ...), l'identification des agents (étiquetés ou non), la nature de mouvements des agents, (synchrone versus asynchrone), le départ des agents simultanément ou avec un délai qui les séparent, la distance initiale qui sépare les deux agents, la capacité de mémoire des agents, le mode de navigation de l'agent (déterministe ou aléatoire), l'utilisation des jetons ou non et finalement la symétrie comme mentionné dans l'article [1] qui distingue la symétrie des agents de celle de l'espace. L'article [1] évoque la symétrie d'agents lorsque les deux agents sont anonymes et ne se distinguent pas l'un de l'autre, par contre la symétrie de l'espace signifie que l'agent ne distingue pas un nœud du graphe ou un point de l'espace d'un autre. Il est à noter que la majorité des articles s'entendent que les agents sont capables de distinguer localement les ports d'un nœud dans lequel ils se trouvent.

1.1 Le modèle

Dans ce mémoire, on s'intéresse au rendez-vous dans une grille, effectué par des agents mobiles ayant des étiquettes différentes. La grille de taille $m \times k$ pour $1 < m \leq k$ est un graphe non-orienté constitué de l'ensemble V de $n = m \times k$ nœuds $v_{i,j}$ avec $1 \leq i \leq m$ et $1 \leq j \leq k$. Chaque nœud $v_{i,j}$ est adjacent aux nœuds $v_{i-1,j}$, $v_{i+1,j}$, $v_{i,j-1}$ et $v_{i,j+1}$ appartenant à l'ensemble V . L'ensemble V est composé de $(m-2)(k-2)$ nœuds de degrés 4 et de $mk - (m-2)(k-2)$ nœuds qui forment le bord de la grille : 4 parmi eux sont de degré 2 et les autres de degré 3. Les agents ne distinguent pas les nœuds de la grille qui ont le même degré, par contre ils distinguent les numéros de ports locaux des nœuds $0, 1, \dots, d-1$, où le nœud v est de degré d . Les nœuds adjacents u et v sont liés par une arête $\{u, v\}$ représentée par les deux numéros de ports des nœuds u et v . Il n'existe aucune relation entre la distribution des numéros de ports du nœud u et v . Lorsqu'un agent visite un nœud, il apprend son degré, le numéro de port par lequel il est entré au nœud courant et peut choisir le numéro de port par lequel il quittera le nœud courant. On n'impose aucune restriction sur la limite de l'espace mémoire des agents. Les agents se déplacent de façon synchrone : dans chaque ronde un agent peut rester dans le même nœud ou aller à un nœud adjacent. Le rendez-vous des agents signifie que les deux agents se trouvent sur un même nœud dans une même ronde. Le temps du rendez-vous c'est le nombre de rondes entre le début du deuxième agent et le rendez-vous. Dans ce mémoire on essaye de résoudre le problème du rendez-vous dans une grille en utilisant les algorithmes suivants :

- marche aléatoire avec retour,
- marche aléatoire sans retour,
- rendez-vous déterministe au bord de la grille,
- rendez-vous déterministe au centre de la grille, le cas général
- rendez-vous déterministe au centre de la grille, lorsqu'il s'agit d'une grille de type échelle.

1.2 Les résultats

On considère deux catégories d'algorithmes, la première aléatoire et la deuxième déterministe. Pour les algorithmes aléatoires, on présente deux variantes, à savoir le

rendez-vous aléatoire avec retour et sans retour. Pour les algorithmes déterministes, on présente deux variantes pour les grilles de taille $m \times k$ pour $1 < m \leq k$, à savoir le rendez-vous déterministe sur le bord de la grille et le rendez-vous déterministe au centre de la grille. Nous présentons deux algorithmes déterministes de rendez-vous pour les grilles $m \times k$, où $2 < m \leq k$: un qui fonctionne en temps $O(2^m + m^2k)$ et l'autre qui fonctionne en temps $O((2^m + k^2)l)$, où l est la plus petite des étiquettes. Le premier algorithme fonctionne pour les grilles des dimensions impairs. Pour les grilles de taille $2 \times k$ on présente un algorithme déterministe qui se réalise en temps de $O(2^k l)$ où l est la plus petite des étiquettes.

Nous faisons des simulations pour estimer le temps du rendez-vous aléatoire.

Chapitre 2

Revue de la littérature

2.1 Les algorithmes de rendez-vous déterministe

Un algorithme de rendez-vous de deux ou plusieurs agents est dit déterministe lorsque les agents exécutent une série de pas qui dépend seulement de l'état de leur mémoire. La plupart des algorithmes déterministes utilisent les étiquettes des agents pour briser la symétrie lors de l'exécution de l'algorithme.

2.1.1 Rendez-vous déterministe synchrone dans un graphe fini

Considérons un graphe connexe de type anneau ayant des nœuds non-étiquetés avec des numéros de ports 1, 2. Deux agents mobiles anonymes étiquetés se déplacent de façon synchrone. Le rendez-vous s'applique lorsque les deux agents arrivent au même nœud au même moment.

L'article [8] exprime le temps du rendez-vous en fonction de la grandeur n de l'anneau, de la distance D entre les deux agents et de la plus petite étiquette l . Pour le délai arbitraire τ des départs des agents, les auteurs présentent deux algorithmes de rendez-vous : un algorithme en temps $O(n \log n)$ pour les anneaux de grandeur connue et un algorithme en temps $O(l\tau + ln^2)$ pour les anneaux de grandeur inconnue. Pour le départ simultané, l'article [8] présente un algorithme qui garantit le rendez-vous dans l'anneau en temps $O(D \log l)$.

De façon générale et pour tout graphe connexe, l'article [8] prouve que pour le départ simultané, un rendez-vous peut être accompli en temps $O(D\Delta^D \log l)$ où Δ est le degré maximal du graphe.

L'article [13] mesure la performance du rendez-vous en fonction du nombre de pas effectués à partir du début de navigation du dernier agent jusqu'au rendez-vous et présente un algorithme RingWalk avec départ non simultané d'ordre $O(n \log l)$ avec n le nombre de nœuds dans le graphe et l la plus petite étiquette des deux agents, alors que pour les graphes généraux il trouve un coût polynomial en n et $\log l$ indépendamment du délai du départ entre les deux agents.

Toujours dans le cas général, l'article [22] utilise la notion Universal Traversal Sequence pour générer une suite de pas synchrones qui garantit le rendez-vous des deux agents après leur activation indépendamment de la topologie du graphe fini ou du temps qui sépare le départ des deux agents. L'article [22] présente une fonction $f(L, d, t)$ pour générer les instructions de navigation basées sur l'étiquette de l'agent L , le degré du nœud courant d et l'unité du temps t à partir du temps de l'activation. L'algorithme garantit une rencontre en temps $O(n^5 \log l)$ après le temps d'activation du deuxième agent indépendamment du délai d'activation entre les deux agents. Avec un graphe de degré d la solution garantit une rencontre après $O(d^2 n^3 \log l)$ pas.

Pour tout graphe de type arbre, le rendez-vous s'effectue au niveau du nœud central ou au niveau des nœuds de l'arête centrale. L'article [8] prouve qu'un rendez-vous des agents étiquetés peut être atteint en temps $O(n + \log l)$, peu importe le nature du départ des deux agents alors que l'article [9] présente un algorithme pour les agents anonymes qui atteint un rendez en un temps optimal de $O(n)$ où n est le nombre de nœuds de l'arbre, pour les positions du départ non-symétriques des agents.

2.1.2 Rendez-vous déterministe asynchrone dans un graphe fini

Les articles [7], [2] et [11] se penchent sur le rendez-vous des agents dans un graphe connexe et infini ou dans le plan. Le mouvement des agents est asynchrone, la vitesse du déplacement est contrôlée par une entité externe arbitraire qui peut diminuer ou augmenter la vitesse de déplacement à condition que le déplacement des agents soit continu.

Vu que le mouvement des agents est asynchrone, le rendez-vous dans un nœud ne peut pas être exigé. Pour cette raison les articles [7],[2] et [11] assouplissent les conditions de rencontre pour accepter comme rendez-vous le scénario où les deux agents se croisent en même temps sur la même arête à condition que les arêtes du graphe ne se croisent pas.

Contrairement à l'article [2], les articles [7] et [11] considèrent le rendez-vous dans les graphes indépendamment de l'étiquetage local des ports des nœuds du graphe. Pour les articles [2] et [7] les agents sont étiquetés et utilisent des algorithmes déterministes pour réaliser le rendez-vous alors que dans le cas de l'article [11] les agents sont identiques et considère une variante aléatoire et une autre déterministe du rendez-vous. Les agents ne connaissent pas la topologie du graphe, et ne peuvent pas laisser de marques sur les nœuds visités.

L'article [7] propose un algorithme GraphRV qui force les deux agents à suivre un parcours qui forme un "tunnel" et ainsi garantit le rendez-vous des deux agents : les agents parcourent respectivement deux chemins R_1 et R_2 ayant comme préfixes respectifs $[e_1, e_2, \dots, e_n]$ et $[e_n, e_{n-1}, \dots, e_1]$. Dans cette situation le rendez-vous est garanti.

À l'aide d'un système de positionnement global qui communique à chaque agent les coordonnées de sa position initiale, l'article [2] propose un algorithme qui garantit le rendez-vous des deux agents dans une grille infinie orientée après avoir parcourus un trajet d'une longueur $O(d^\sigma \text{polylog } d)$ où σ représente la dimension de la grille et d représente la distance initiale qui sépare les deux agents. Chaque agent ne connaît pas la position initiale de l'autre agent.

L'article [2] prouve que pour une grille de dimension σ , il existe une configuration qui optimise la distribution des numéros de ports locaux des nœuds nécessitant $O(2^\sigma)$ différentes étiquettes assignées aux différents nœuds du graphe. Cette configuration est capable de donner assez d'informations aux agents pour garantir un rendez-vous après avoir parcouru une trajectoire de longueur $O(d^\sigma \log^{\sigma^2 + \sigma + 1} d)$.

L'article [11] combine la vue de l'agent avec la notion du chemin palindrome pour garantir le rendez-vous des deux agents anonymes. Un chemin r fini est dit palindrome, si le chemin r et son chemin inverse \bar{r} sont induits par la même séquence de numéros de ports. En d'autres termes, la séquence des ports lus lors du parcours du trajet r est la même que lors du parcours du trajet \bar{r} . Dans le cas déterministe, l'article [11] présente un algorithme Deterministic-RV qui garantit le rendez-vous si les vues à partir des positions initiales des agents sont différentes ou bien si les positions initiales sont connectées par un palindrome. L'article [11] prouve que dans le cas où les deux conditions ne sont pas vérifiées le rendez-vous ne peut pas être garanti. Dans le cas aléatoire, l'article [11] présente l'algorithme Randomized-RV qui garantit le rendez-vous avec une probabilité égale à 1 pour des positions initiales arbitraires. Dans le cas du plan, l'article [11] présente une variante de l'algorithme Randomized-RV utilisant une boussole, qui garantit avec

une probabilité égale à 1 de ramener les deux agents à une distance connue par les deux agents égale à ϵ avec $\epsilon > 0$ arbitraire.

2.2 Les algorithmes de rendez-vous aléatoires

Un algorithme de rendez-vous est dit aléatoire lorsque les agents utilisent une génératrice de nombres aléatoires pour déterminer leurs mouvements. Ceci implique que les deux agents ne suivent pas nécessairement la même suite de nœuds lorsqu'ils sont dans les mêmes circonstances.

Pour un réseau de type anneau, l'article [12] distingue entre les algorithmes avec détection et les algorithmes sans détection. Ainsi, l'article [12] qualifie un rendez-vous avec détection, tout algorithme qui a une exécution finie dans le temps et qu'à l'état final du rendez-vous tous les agents se retrouvent dans un seul nœud du graphe et préservent leur état final '*arrêt*' (*halt*), alors qu'un rendez-vous sans détection permet aux agents de préserver l'état '*suspendu*' qui ne permet pas à l'agent de déduire la fin du programme sachant qu'un autre agent peut visiter son nœud à n'importe quel moment et modifier l'état du nœud.

On dit qu'un algorithme est capable d'atteindre un rendez-vous avec une probabilité supérieure ou égale à p , si pour toute configuration initiale C_0 le rendez-vous se réalise avec une probabilité supérieure ou égale à p .

L'article [12] considère le problème de rendez-vous pour k agents avec ($k \geq 2$) dans un réseau en anneau unidirectionnel et suppose que les agents ne connaissent ni le nombre de nœuds ni le nombre d'agents dans l'anneau. L'article [12] prouve que pour toute constante p , telle que $0 < p \leq 1$, il n'existe aucun algorithme aléatoire avec détection capable de résoudre le problème de rendez-vous avec une probabilité p . Ceci explique le choix de résolution du problème de rendez-vous avec l'option sans détection. L'article [12] prouve qu'il n'existe aucun algorithme aléatoire capable de résoudre le rendez-vous avec une probabilité égale à 1. Par contre, l'article propose un algorithme qui résout pour chaque probabilité constante p , ($0 < p < 1$), le problème de rendez-vous sans détection.

L'article [16] considère le temps de rendez-vous des agents modélisés comme des automates finis. Il présente un algorithme qui peut atteindre un rendez-vous dans un délai de $O(n^2/2^t + 2^t)$ utilisant un automate fini de $2t$ états alors que pour un automate à $t/2$ états le rendez-vous exige un temps $\Omega(n^2/2^t)$.

Pour la structure de l'arbre, l'article [9] utilise un algorithme aléatoire pour accélérer le rendez-vous. Ses auteurs présentent un algorithme qui atteint le rendez-vous en temps $O(\log n)$, avec forte probabilité, si le degré maximal est constant. Par contre l'algorithme prend au moins un temps de $\Omega(n)$ avant d'atteindre le rendez-vous même si la distance initiale qui sépare les deux agents est égale à 1, si le degré maximal peut être non-borné.

2.3 Les algorithmes de rendez-vous utilisant des jetons

L'article [15] est l'un des premiers articles à établir une relation entre le nombre de jetons utilisés et l'espace mémoire nécessaire aux agents identiques pour atteindre un rendez-vous dans un torus $n \times m$ synchrone en utilisant un algorithme déterministe. L'article prouve qu'il y a un lien fort entre le nombre de jetons utilisés et l'espace mémoire des agents. Ainsi : deux agents avec un nombre constant de jetons stationnaires ou bien un seul jeton amovible, ne sont pas capables de se rencontrer s'ils ont une capacité mémoire égale à $o(\log n)$; avec un seul jeton stationnaire les mêmes agents peuvent réaliser un rendez-vous avec une capacité mémoire égale à $O(\log n)$ même en utilisant un rendez-vous avec détection. Deux agents munis de deux ou trois jetons stationnaires, peuvent réaliser le rendez-vous avec un espace mémoire constant s'ils utilisent un algorithme de rendez-vous avec détection. L'article [17] considère un réseau moins complexe que l'article [15], à savoir l'anneau et prouve que pour une distance de $d = n/2$ où n est le nombre de nœuds dans le réseau, il est impossible de trouver un algorithme qui garantit le rendez-vous en utilisant des jetons stationnaires. Avec cette limite, l'article [17] considère le rendez-vous de deux agents synchrones ayant chacun un jeton stationnaire. Comme illustré dans la table 2.1, l'article [17] présente une série de résultats concourant la capacité mémoire nécessaire pour le rendez-vous et le temps de rendez-vous, en fonction de la connaissance de nombre de nœuds du graphe, la distance initiale et l'orientation du graphe.

TABLE 2.1 – Temps du rendez-vous en fonction du nombre de nœuds, de l'orientation de l'anneau et de la distance initiale qui sépare les agents

Nombre de nœuds connus	Distance initiale connue	Orientation de l'anneau connue	Limite Inf./Limite Sup.	Mémoire requise
Oui	Oui	Oui	$3d/2$	$O(\log d)$
Oui	Oui	Non	$3d/2$ $5d/2$	$O(\log d)$
Oui	Non	Oui	$(n + d)/2$	$O(\log n)$
Oui	Non	Non	$(n + d)/2$	$O(\log n)$
Non	Oui	Oui	$3d/2$	$O(\log d)$
Non	Oui	Non	$3d/2$ $5d/2$	$O(\log d)$
Non	Non	Oui	$n + (d/2)$	$O(\log n)$
Non	Non	Non	$n + (d/2)$	$O(\log n)$

Toujours dans le cadre des anneaux, l'article [10] étudie le temps nécessaire pour arriver au rendez-vous lorsqu'un ou plusieurs jetons sont perdus lors de l'exploration. L'article [10] suppose que les agents connaissent le nombre de jetons k ou/et le nombre d'agents n présents dans l'anneau. L'article [10] établit le temps du rendez-vous, comme illustré dans la table 2.2, dépendamment du temps de la perte des jetons, soit au début de l'exploration soit à n'importe quel moment. Les résultats se résument comme suit :

TABLE 2.2 – Temps du rendez-vous en fonction du moment de perte des jetons

Perte de jeton	Connaissance	Temps
Jamais	n ou k	$O(n)$
	n	$O(n)$
Au début	k	$O(kn)$
	n	$O(kn)$
N'importe quand	n	$O(kn)$
	k	$O(k2n)$

Chapitre 3

Les algorithmes

Dans cette section nous décrivons nos algorithmes de rendez-vous.

3.1 Les algorithmes aléatoires

3.1.1 La marche aléatoire avec retour jusqu'au rendez-vous

À partir de la position initiale, l'agent choisit aléatoirement un numéro de port parmi les ports du nœud courant pour se déplacer vers l'un des nœuds adjacents. L'agent utilise une fonction aléatoire uniforme pour générer un entier p avec $0 \leq p \leq d$ où d est le degré du nœud courant. Si le résultat est d , l'agent reste sur place, si le résultat est $0 \leq p < d$, l'agent part par le port p . L'action est répétée jusqu'à ce que le rendez-vous se réalise. Le déplacement de l'agent à l'instant t_{i+1} est complètement indépendant du déplacement effectué à l'instant t_i , autrement dit, l'agent ne se souvient pas de l'historique de son parcours et peut à l'instant t_{i+1} retourner à la position dont il est parti à l'instant t_i . Les simulations nous aideront à établir le temps de fonctionnement de l'algorithme.

3.1.2 La marche aléatoire sans retour jusqu'au rendez-vous

L'algorithme aléatoire sans retour est le même que celui avec retour à l'exception que l'agent à l'instant t_{i+1} ne recule pas vers le nœud dont il est parti à l'instant t_i , c'est à dire, les numéros de ports choisis aléatoirement excluent le dernier port d'entrée.

3.2 Les algorithmes déterministes

3.2.1 Boîte à outils

Dans cette section on présente quelques procédures que nous utiliserons dans les algorithmes déterministes.

La procédure **repos-action** de rendez-vous sur un anneau [8] sera utilisée pour réaliser le rendez-vous de deux agents sur le bord d'une grille, d'un carré ou sur une arrête. L'agent effectue une série d'actions alternatives, repos pour KL rondes et marche dans la même direction pour KL rondes, avec $K = 1, 2, 3 \dots$ et L l'étiquette de l'agent. La rencontre se réalise dans l'ordre de $O(l\tau + ln^2)$ avec l la plus petite étiquette des deux agents, τ le délai qui sépare le départ de deux agents et n la grandeur de l'anneau.

Pour $K = 1, 2, \dots$

- Bouge KL étapes dans une direction arbitraire.*
- Reste stationnaire KL étapes*

L'article [13] présente une variante plus performante que la procédure **repos-action**, appelée **RingWalk** qui réalise la rencontre avec délai arbitraire en un temps de $O(n \log l)$ avec l la plus petite des deux étiquettes. Cependant, **RingWalk** est plus difficile à implémenter. Le gain en performance entre les deux variantes est négligable vu que la procédure sera utilisée en conjonction avec des procédures d'ordre exponentiel, ce qui explique notre choix pour la variante la plus couteuse mais, la plus facile à implémenter à savoir, la procédure **repos-action**.

La procédure **Classification de grilles**.

On distingue trois classes de grilles A, B et C . qui se définies comme suit :

- classe A** : grilles $m \times k; m, k = 2$
- classe B** : grilles $m \times k; m = 2, k > 2$ (échelle)
- classe C** : grilles $m \times k; m, k > 2$

L'agent commence à un nœud donné v . Si v est de degré 4, la procédure déclare la grille de **classe C**. Si v est de degré 3, l'agent visite les nœuds voisins, si au moins l'un des voisins est de degré 4, la grille est déclarée de **classe C**, sinon elle est de classe **B**. Si v est de degré 2, l'agent visite tous les nœuds voisins. S'ils sont tous de degrés 2, alors

la grille est de **classe A**, si exactement un des voisins est de degré 2, alors la grille est de **classe B**, sinon elle est de **classe C**.

Le temps de fonctionnement de la procédure est constant.

Procédure Recherche exponentielle du bord de la grille

Cette procédure s'applique aux grilles de **classe C**. Elle est utilisée par un agent mobile pour atteindre le bord de la grille à partir de sa position courante. L'état final de la procédure est réalisée lorsque l'agent visite un nœud de degrés 3 ou 2. L'agent commence par évaluer le degré du nœud courant v . Si le nœud est de degré 3 ou 2, la procédure atteint son état final. Sinon, pendant la phase $d = 1, 2, 3, \dots$, l'agent visite tous les nœuds qui peuvent être atteint à partir du nœud de départ par des chemins de longueur d . Si l'un des nœuds visités est de degré 3 ou 2, la procédure atteint l'état final, sinon l'agent recule à sa position initiale avant de continuer l'exploration. Le coût de cette procédure est de $O(2^m)$, avec d la distance qui sépare l'agent du bord de la grille.

Remarque :

Dans l'article [20] l'auteur montre un algorithme qui permet d'explorer n'importe quel graphe en temps polynomial dans sa grandeur. Cet algorithme appliqué à la grille permet en particulier de trouver un nœud du bord de la grille en temps polynomial. Cependant cet algorithme est très compliqué et le polynôme est de très haut degré, donc au lieu de l'utiliser, nous appliquons la procédure exponentielle ci-dessus qui pour un d de grandeur modérée donne une efficacité acceptable.

Procédure Exploration du bord de la grille

Cette procédure s'applique aux grilles de **classe C**. La procédure effectue un tour complet du bord de la grille et mémorise la séquence des ports qui part de la position courante de l'agent sur le bord de la grille et finit à la position initiale de l'agent. À partir du nœud courant v , l'agent visite le premier nœud voisin w à travers le port 0. Si w est d'ordre 4, l'agent recule d'un pas. Le prochain voisin x de v visité à travers le port 1 ou 2, doit certainement être d'ordre 2 ou 3. Ainsi, après au plus 3 pas, l'agent visite un nœud voisin de degré 2 ou 3. Soit v_0 , la position initiale de l'agent. Le parcours de la grille s'arrête après avoir visité les quatre nœuds de degré 2 de la grille, respectivement C_1, C_2, C_3 et C_4 plus la distance $Distance(C_4, v_0)$ qui est calculée comme $Distance(C_4, v_0) = Distance(C_2, C_3) - Distance(v_0, C_1)$.

Le temps de cette procédure est égal à $O(m + k) = O(k)$ pour les grilles $m \times k$ avec $2 < m \leq k$.

3.2.2 Algorithme : Rencontre sur le bord de la grille

Cet algorithme s'applique sur les grilles de classe A et C. On applique d'abord la procédure **Classification de grilles**. Pour les grilles de classe C, la rencontre commence lorsque l'agent atteint le bord de la grille en utilisant la procédure **Recherche exponentielle du bord de la grille** et reconnaît le bord en utilisant la procédure **Exploration du bord de la grille**. Une fois les deux procédures terminées, l'agent commence à exécuter la procédure **repos-action** sur le bord de la grille considérée comme anneau. La procédure **repos-action** réalise le rendez-vous en un temps de $O(l\tau + lx^2)$, où $x = 2m + 2k$ est la longueur du bord de la grille, τ est la différence du temps du début de la procédure **repos-action** entre les agents et l la plus petite étiquette. Puisque la différence τ est de l'ordre $O(4^{m/2}) = O(2^m)$ et $x = 2m + 2k$ est de l'ordre de $O(k)$, le temps total de l'exécution de l'algorithme est de l'ordre de $O(l2^m + lk)$, où l est la plus petite des étiquettes.

Dans le cas des grilles de type A, la procédure **repos-action** est lancée tout de suite après la classification de la grille. Dans ce cas, elle réalise le rendez-vous en temps $O(l)$ puisque τ est constant et la grandeur de l'anneau est 4.

3.2.3 Algorithme : Rencontre au centre de la grille : Classe C

Cet algorithme s'applique aux grilles de classe C ayant des valeurs impaires pour les deux dimensions de la grille. La rencontre au centre de la grille commence après la classification des grilles, la recherche exponentielle du bord de la grille et la reconnaissance du bord de la grille. Cette procédure est divisée en trois sous procédures, à savoir, la procédure **d'exploration à partir du coin de la grille**, la procédure **compléter le quart** et la procédure **de détermination du centre de la grille**.

La procédure de **l'exploration à partir du coin de la grille** est utilisée pour les grilles de taille $m \times k$, avec $2 < m \leq k$. Au début de la procédure, l'agent est positionné sur le coin C de la grille. Soit $x = \lceil m/2 \rceil - 2$ et $y = \lceil k/2 \rceil - 2$. Soit l'axe horizontal de la grille celui qui commence au coin C de la grille et de longueur k et l'axe vertical celui qui commence au coin C de la grille et de longueur m . Si $m = k$, le choix des noms des axes est fait de façon arbitraire.

Soit v_i , avec $1 \leq i \leq y$, le nœud situé sur l'axe horizontal à distance i du coin C de la grille et soit w_i , avec $1 \leq i \leq x$, le nœud situé sur l'axe vertical à distance i du coin C de

la grille. Soit la i ème ligne horizontale qui commence au nœud w_i l'ensemble des nœuds horizontaux à distance maximale y de w_i . Soit la i ème colonne verticale qui commence au nœud v_i l'ensemble des nœuds verticaux à distance maximale x de v_i . Soit Q_s avec $1 \leq s \leq x - 1$, l'union des nœuds des premières s lignes et s colonnes. L'objectif de cette procédure est de construire une carte partielle de tous les nœuds appartenant à Q_{x-1} . La procédure se termine après $x - 1$ étapes. À l'étape $s = 1, 2, \dots, x - 1$ les nœuds de la ligne s et la colonne s sont ajoutés à la carte partielle Q_s . La procédure se fait en s étapes, avec $s = 1, 2, \dots, x - 1$. Au début de l'étape s , l'agent connaît la carte partielle de tous les nœuds appartenant au bord de la grille plus les nœuds appartenant à l'ensemble Q_{s-1} . Nous décrivons l'étape s .

Soit le nœud g à distance $s - 1$ des nœuds v_{s-1} et w_{s-1} . Soit z_j avec $s \leq j \leq y$, le nœud à distance $s - 1$ du nœud v_j et à distance j du nœud w_{s-1} . Soit t_j avec $s \leq j \leq x$, le nœud à distance $s - 1$ du nœud w_j et à distance j du nœud v_{s-1} . Soit π_j la séquence des ports du bord de la grille qui part du nœud v_j et qui finit au coin C de la grille, soit ρ_j la séquence des ports du bord de la grille qui part du nœud w_j et qui finit au coin C de la grille, soit π'_j la séquence des ports sur le chemin horizontal qui part du nœud t_j et qui finit au nœud w_j et soit ρ'_j la séquence des ports sur la ligne verticale qui part du nœud z_j et qui finit au nœud v_j comme illustré dans la figure 3.1.

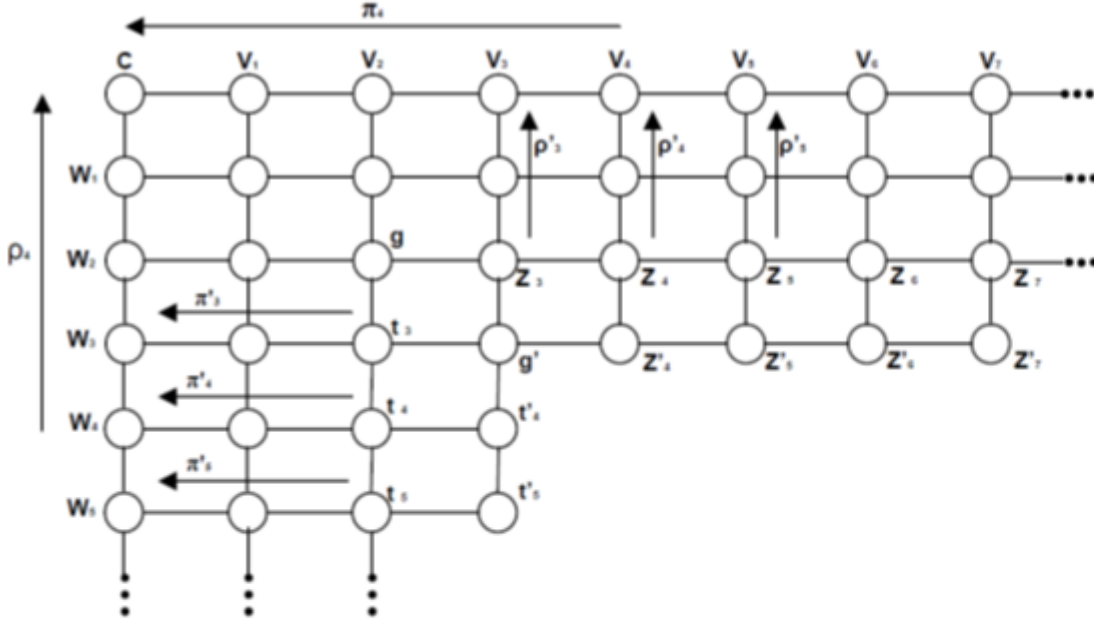


FIGURE 3.1 – La procédure d’exploration à partir du coin de la grille à l’étape $s = 3$.

L’agent se positionne sur le nœud z_s et sélectionne le port q' pas encore enregistré dans la carte Q_{s-1} et se rend au nœud g' . L’agent prend le chemin $z_s g t_s$ pour se rendre au nœud t_s et à partir de t_s se rend au nœud g' en prenant le port q'' qui n’est pas encore enregistré dans la carte Q_{s-1} . L’agent sélectionne l’un des deux ports non visités du nœud g' , soit r' , et se rend au nœud adjacent h' . Ce dernier peut être soit le nœud voisin horizontal z'_{s+1} ou bien le voisin vertical t'_{s+1} du nœud g' . L’agent visite tous les voisins du nœud h' et à partir de chacun d’entre eux il prend le chemin π'_{s+1} . Après chaque essai, l’agent revient à sa position initiale h' . Au plus un des essais permet à l’agent de se rendre au bord de la grille au bout du chemin π'_{s+1} . Considérons les deux cas suivants :

Cas 1 : à partir de l’un des voisins du nœud h' , le chemin π'_{s+1} mène vers un nœud qui se trouve sur le bord de la grille, soit w_{s+1} ou bien v_{s+1} . L’agent identifie le nœud qui se trouve sur le bord de la grille en prenant le chemin ρ_{s+1} . Si l’agent atteint le coin C de la grille alors le nœud est w_{s+1} sinon c’est v_{s+1} . Dans le premier cas le nœud h' était le voisin vertical t'_{s+1} du nœud g' et le chemin π'_{s+1} avait commencé à partir du nœud t_{s+1} .

Dans le deuxième cas le nœud h' était le voisin horizontal z'_{s+1} du nœud g' et le chemin π'_{s+1} avait commencé à partir du nœud z_{s+1} (ceci est possible si $\pi'_{s+1} = \rho'_{s+1}$). L'agent ajoute le nœud h' à la carte partielle Q_{s-1} et retourne au nœud g' .

Cas 2 : pour tous les voisins du nœud h' , le chemin π'_{s+1} ne mène pas vers le bord de la grille. Cela veut dire que h' est le voisin horizontal z'_{s+1} du nœud g' . L'agent visite tous les voisins du nœud h' et essaye d'arriver au bord de la grille en prenant le chemin ρ'_{s+1} . Après chaque essai, l'agent revient au nœud h' . Parmi tous les voisins de h' , il existe un seul port r^* qui mène au nœud v_{s+1} au bout du chemin ρ'_{s+1} . L'agent ajoute le nœud h' à la carte partielle Q_{s-1} et retourne au nœud g' . Dans les deux cas, l'agent a ajouté un seul nœud à la carte partielle. L'agent sélectionne le dernier port non visité r'' du nœud g' et visite le nœud h'' . Si $h' = z'_{s+1}$ alors $h'' = t'_{s+1}$ et si $h' = t'_{s+1}$ alors $h'' = z'_{s+1}$. L'agent enregistre le nœud h'' dans la carte partielle. Il reste à enregistrer le port approprié. Dans le cas où $h'' = z'_{s+1}$, l'agent visite tous les voisins du nœud h'' et à partir de chacun d'eux prend le chemin ρ'_{s+1} à chaque fois en revenant au point de départ h'' . Il existe un seul voisin du nœud h'' qui mène au nœud v_{s+1} à partir du nœud z_{s+1} en prenant le chemin ρ'_{s+1} . L'agent enregistre le port approprié et retourne au nœud g' . Dans le cas où $h'' = t'_{s+1}$, l'agent visite tous les voisins du nœud h'' et à partir de chacun d'eux il prend le chemin π'_{s+1} à chaque fois en revenant au point de départ h'' . Il existe un seul voisin du nœud h'' qui mène au nœud w_{s+1} à partir du nœud t_{s+1} en prenant le chemin π'_{s+1} . L'agent enregistre le port approprié et retourne au nœud g' .

En ce moment la carte partielle contient les nœuds g' , z'_{s+1} et t'_{s+1} . Le reste de l'étape s est divisé en deux parties. Dans la première partie, le reste de la s -ième ligne est ajouté à la carte partielle et pour la deuxième partie, le reste de la s -ième colonne est ajouté à la carte partielle. Juste la première partie est détaillée, la deuxième s'effectue de la même façon.

La première partie se fait en plusieurs étapes, à chaque étape, un nœud est ajouté à la ligne avec les numéros de ports appropriés. Supposons que le nœud z'_i avec $i \leq j$ qui se trouve sur la ligne s a été ajouté, l'agent se trouve sur le nœud z'_j comme présenté dans la figure 3.1. Le nœud z'_j compte deux ports qui n'ont pas encore été visités. L'agent prend tous les chemins de longueur 2 à partir des deux ports non visités du nœud z'_j et suit le chemin ρ'_{j+1} . Parmi tous les chemins possibles, il existe un seul qui mène vers le bord de la grille, plus exactement au nœud v_{j+1} . Le voisin correspondant de z'_j est z'_{j+1} . L'agent enregistre ce nœud et le port approprié et revient au nœud z'_{j+1} . Ceci complète cette étape de la première partie.

Après avoir terminé les deux parties de l'étape s , la ligne et la colonne qui se trouvent à l'indice s sont ajoutés à la carte partielle. L'agent se repositionne au nœud g' et il est prêt à entamer l'étape $s + 1$. Ceci complète la description de la procédure.

À la fin de la **procédure d'exploration à partir du coin de la grille** l'agent mémorise tous les nœuds qui appartiennent à l'union Q_{x-1} avec les numéros des ports des arêtes qui les lient. Le temps de l'ajout des trois premiers nœuds à chaque étape s est toujours de $O(m)$. Chaque pas de l'étape s a un temps de $O(m)$. Vu qu'il y a au total $O(mk)$ pas (chaque pas ajoute un nœud), le temps total de la procédure est de $O(m^2k)$.

La procédure **compléter le quart**, ajoute à la carte partielle le reste de la x -ième ligne, pour former la sous-grille complète Q_{x-1} . Ceci est fait de façon semblable comme dans la procédure d'exploration à partir du coin de la grille et prend un temps $O(mk)$.

Pour toute grille de taille $m \times k$, où m et k sont impairs, avec $2 < m \leq k$, soit k la longueur des lignes horizontales appelées lignes et m la longueur des lignes verticales appelées colonnes. Supposons que les procédures **exploration à partir du coin de la grille** et **compléter le quart** ont été exécutées pour les quatre coins de la grille. L'union des ensembles visités Q_{x-1} couvrira toute la grille excepté la ligne médiane et la colonne médiane comme le montre la figure 3.2.

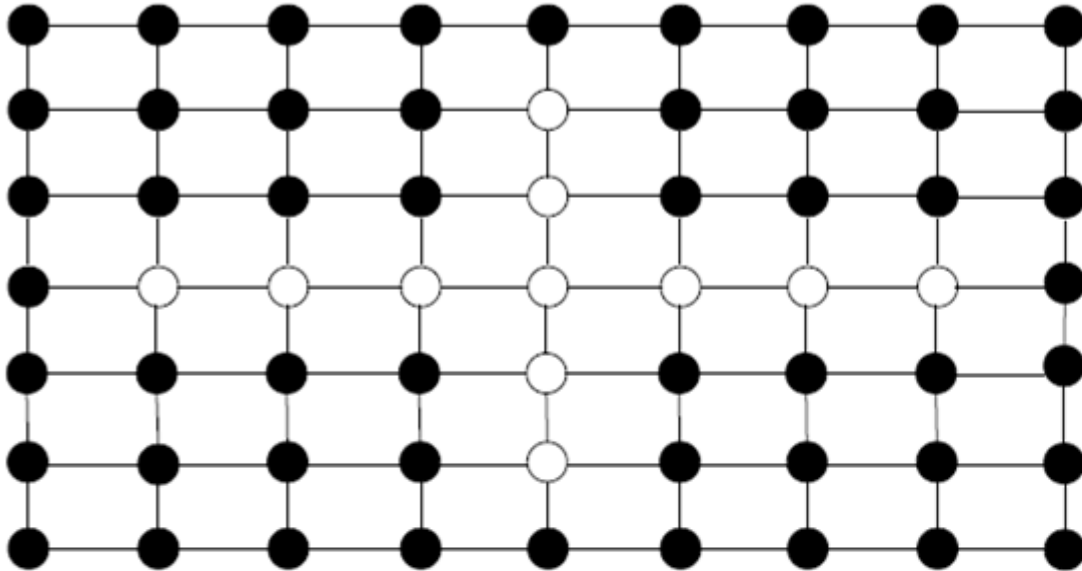


FIGURE 3.2 – Une grille de taille 7×9 après avoir exécuté la procédure `exploration` à partir du coin de la grille et la procédure `compléter le quart` pour les quatre coins de la grille. Les nœuds marqués sur la carte partielle sont noirs.

La procédure qui suit est celle de la `détermination du centre de la grille`. La procédure commence à partir de l'un des deux nœuds qui se trouvent sur le bord de la grille et sur la colonne médiane. L'objectif de cette procédure est de construire la suite des numéros de ports qui mène vers le centre de la grille. Une fois positionné sur le centre v du côté horizontal du bord de la grille qui se trouve à distance y du coin de la grille avec $y = \lceil k/2 \rceil - 2$, l'agent commence à tracer la séquence de ports sur le chemin vertical qui mène vers le centre de la grille. La procédure se fait en s étapes avec $s = 1, 2, \dots, x$ et $x = \lceil m/2 \rceil - 1$. Pour $s = 1$, l'agent connaît le port qui le mène de v vers le voisin vertical.

Soit b_s , pour $s = 1, 2, \dots, x$, le nœud à distance s de v sur la colonne centrale. Soient a_s et c_s les deux voisins horizontaux de b_s . À la fin de l'étape s l'agent est au nœud b_s . Soit π_s le chemin vertical (codé comme une suite de ports) du nœud a_s au voisin horizontal de v et soit ρ_s le chemin vertical du nœud c_s au voisin horizontal de v . À l'étape $s + 1$, pour $s = 1, 2, \dots, x - 1$, l'agent doit arriver au nœud b_{s+1} . Au nœud b_s , il y a trois ports qui ne sont pas encore utilisés. Deux de ces ports mènent vers les nœuds a_s et c_s et un port mène vers b_{s+1} . L'agent prend chacun des trois ports et ajoute les chemins π_s et ρ_s (qui sont déjà sur sa carte partielle). Pour exactement deux des ports essayés l'agent

se trouve au bord de la grille au bout d'un de ces chemins. Après chaque essai l'agent revient à b_s . Le troisième port mène vers b_{s+1} , ce qui termine l'étape $s + 1$. À la fin de l'étape x l'agent est au centre de la grille et s'arrête.

Le temps de chaque étape est de $O(m)$, donc le temps total de la procédure est de $O(m^2)$. Puisque la procédure Recherche exponentielle du bord de la grille prend un temps $O(2^m)$, la procédure Reconnaissance du bord de la grille prend un temps $O(k)$, la procédure Exploration à partir du coin de la grille prend un temps $O(m^2k)$, la procédure Compléter le quart prend un temps $O(mk)$ et la procédure Détermination du centre de la grille prend un temps $O(m^2)$, le temps total de l'algorithme est de $O(2^m + m^2k)$.

3.2.4 Algorithme : Rencontre au centre de la grille : Classe B

L'algorithme commence par la procédure classification des grilles et s'applique aux grilles de class B de format $2 \times k$. Soit le nœud t ayant les trois ports p_1 , p_2 et p_3 , la position de départ de l'agent comme illustré dans la figure 3.3.

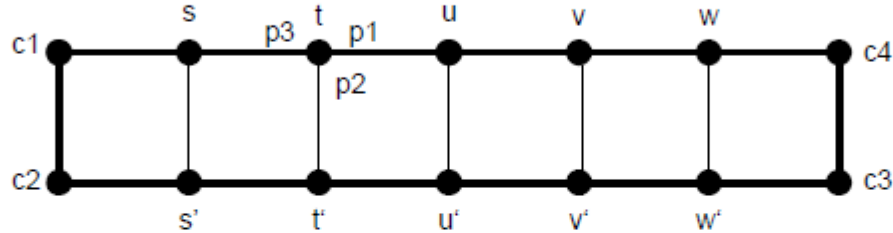


FIGURE 3.3 – Échelle de taille 2×7

À l'étape $e = 1$, l'agent explore tous les nœuds voisins du nœud courant t à travers les trois ports p_1 , p_2 et p_3 . À la fin de l'étape $e = 1$ l'agent construit un arbre de profondeur $e = 1$ ayant comme racine le nœud t et comme feuilles les nœuds u , t' et s .

Après la fin de chaque étape, l'agent vérifie s'il a atteint un nœud de degré 2.

Cas 1 : À la fin de l'étape e , l'agent n'arrive pas à un nœud de degré 2. Dans ce cas, l'agent procède à l'étape $e + 1$, et ajoute tous les nœuds voisins des nœuds à profondeur

e de l'arbre. À la fin de l'étape $e = 2$, l'agent ajoutera au niveau $e = 2$ de l'arbre les nœuds suivants :

- c_1 et s' , nœuds voisins du nœud s
- s' et u' , nœuds voisins du nœud t'
- u' et v , nœuds voisins du nœud u

Si à la fin de l'étape $e = 2$, l'agent ne trouve pas un nœud de degré 2, il procède aux étapes suivantes $e = 3, 4, 5, \dots$ jusqu'à ce qu'il trouve un coin parmi les quatre coins de la grille. Dans notre cas, à l'étape $e = 2$, l'agent trouve un nœud de degré 2 (c_1), ce qui nous amène au **Cas 2**.

Cas 2 : À la fin de l'étape e , l'agent trouve un ou plusieurs nœuds de degré 2. L'agent calcule les caractéristiques de tous les nœuds de l'arbre. La caractéristique d'un nœud donné v qui correspond à un port p est un chemin de longueur minimale du nœud v à un coin, partant par le port p . Chaque nœud peut avoir une ou deux caractéristiques correspondant à chaque port. L'algorithme cherche les deux ou quatre nœuds centre de l'échelle qui sont les seuls à avoir les caractéristiques suivantes :

- Pour une échelle $2 \times k$ avec k impair (ayant comme centre une arête) : chacun des deux nœuds au centre de la grille, aura deux caractéristiques de longueur x correspondant à deux ports et deux caractéristique de longueur $x+1$ correspondant au troisième port.
- Pour une échelle $2 \times k$ avec k impair (ayant comme centre un carré) : chacun des quatre nœuds du carré central de l'échelle aura deux caractéristiques de longueur $x + 1$ correspondant à deux ports et une caractéristique de longueur x correspondant au troisième port.

L'algorithme continue la construction de l'arbre des chemins de profondeur $e = 1, 2, \dots$, jusqu'à ce que une des conditions ci-dessus soit satisfaite. Les 2 ou 4 nœuds qui satisfont cette condition forment l'arête centrale ou le carré central.

Voici la liste des caractéristiques à la fin de l'étape $e = 2$:

TABLE 3.1 – la liste des caractéristiques à la fin de l'étape $e = 2$

Caractéristiques	Longueur de la caractéristique
t, s, c_1	2
s, c_1	1
s', s, c_1	2
t', t, s, c_1	3
s', t', t, s, c_1	4
u', t', t, s, c_1	4
u, t, s, c_1	3
u', u, t, s, c_1	4
v, u, t, s, c_1	4

Comme le démontre la table 3.1, aucun nœud ne répond aux critères du potentiel nœud centre (3 caractéristiques qui partent du même nœud de trois ports différents de longueurs x , x , et $x + 1$,).

Si l'agent ne trouve pas le centre de l'échelle, il passe à l'étape $e = 3$ et ajoute tous les nœuds voisins des nœuds de l'arbre qui se trouve à la profondeur $e = 2$. L'étape est répétée pour $e = 4, 5, \dots$, jusqu'à ce que le centre soit trouvé.

À l'étape $e = 5$, l'agent trouve le centre du graphe u ayant les trois caractéristiques suivantes :

TABLE 3.2 – les caractéristiques de centre de l'échelle

Caractéristiques	Longueur de la caractéristique
u, t, s, c_1	3
u, u', t', s', c_2	4
u, v, w, c_4	3

Au moment de la découverte d'un nœud centre de l'échelle on trouve aussi l'arête ou le carré central qui est formé de nœuds (voisins ou à distance 2) ayant les mêmes caractéristiques que celle du premier nœud centre trouvé.

La recherche du centre de l'échelle est donc composée de deux étapes :

- Construction de l'arbre des chemins jusqu'à ce que l'agent trouve un nœud de degré 2.
- La recherche des caractéristiques jusqu'à la satisfaction de une des deux conditions indiquant le centre.

Une fois le centre trouvé, l'agent fait appel à la procédure **repos-action** sur l'arrête centrale ou sur le carré central. La recherche du centre se fait en construisant un arbre de chemins de longueurs au plus k ce qui prend un temps $O(2^k)$. Ensuite la procédure **repos-action** prend un temps $O(l\tau)$, puisqu'elle est appliquée sur un anneau de grandeur constante, où τ est le délai de la fin de recherche du centre entre les deux agents. Puisque τ est de l'ordre de $O(2^k)$, le temps total de l'algorithme est de $O(l2^k)$, où l est la plus petite étiquette.

Chapitre 4

La simulation

Dans cette section, on présentera les résultats des simulations des algorithmes décrits dans le chapitre précédent. La simulation a été implantée dans le langage de programmation C# en utilisant la librairie `.NET Framework 4.0` sur une machine Intel à 2 processeurs de vitesse 2.4 GHZ ayant trois mille octet de mémoire virtuelle.

4.1 Résultats pour la simulation du rendez-vous au bord d'une grille de taille $m \times k$, avec $2 < m \leq k$

4.1.1 Données d'entrée

Pour la Rencontre au bord de la grille on considère huit tailles de grilles différentes :

- 5×10
- 5×20
- 5×50
- 5×100
- 10×10
- 20×20
- 50×50
- 100×100

Pour chaque taille on considère les cinq configurations de départs suivantes :

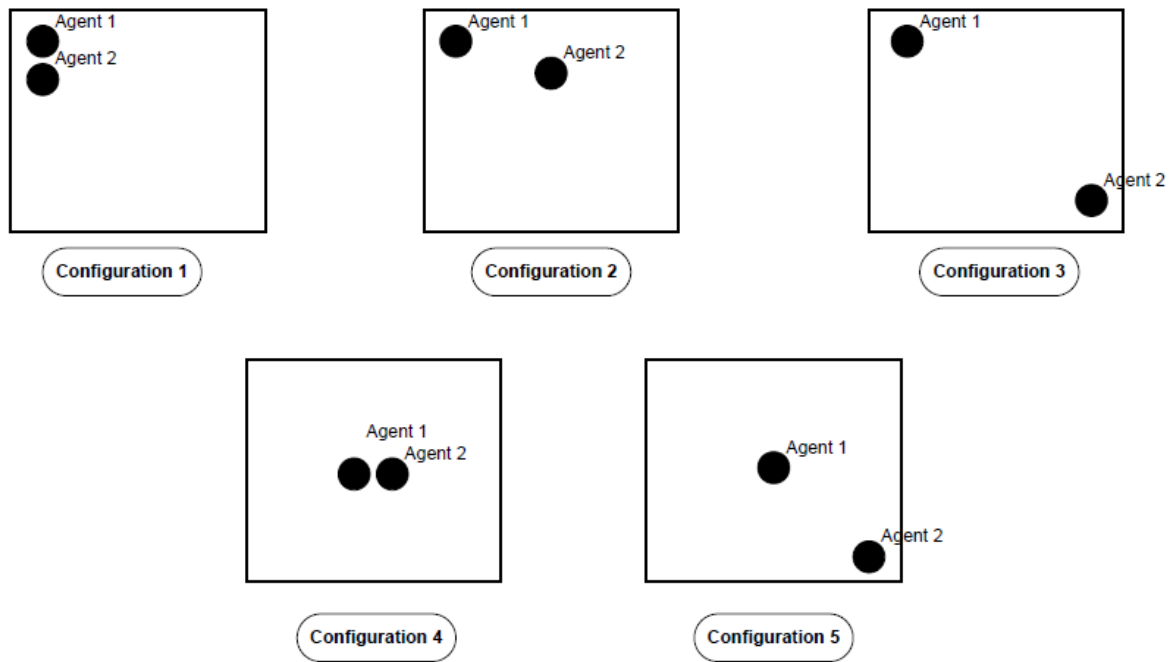


FIGURE 4.1 – Les cinq configurations de départ pour le rendez-vous au bord de la grille

- Pour la configuration 1, on s’attend à ce que les agents arrivent relativement en même temps au bord de la grille.
- Pour la configuration 2, on s’attend à ce que l’agent 2 prenne plus de temps pour arriver au bord de la grille alors que l’agent 1 fera plus de pas autour de la grille avant de réaliser le rendez-vous.
- Pour la configuration 3, on s’attend à ce que les agents arrivent en même temps au bord de la grille et mettent plus de temps pour réaliser le rendez-vous.
- Pour la configuration 4, on s’attend à ce que les deux agents arrivent en même temps au bord de la grille mais, en faisant beaucoup plus de pas que pour les autres configurations.
- Pour la configuration 5, on s’attend à ce que l’agent 1 arrive plus tard au bord de la grille et que l’agent 2 effectue plus de pas avant de réaliser le rendez-vous

Pour chaque taille de grille et pour chaque configuration de départ, l’algorithme est exécuté 100 fois. Pour chaque exécution, le graphe est réinitialisé en redistribuant de

façon aléatoire les numéros de ports sur les nœuds de la grille.

Pour la rencontre au bord d'une grille de taille 100×100 , la configuration 4 a été omise vu que la recherche exponentielle du bord de la grille prendra beaucoup de temps et de ressource pour s'exécuter. Pour le reste des tailles, on a fixé la distance maximale entre la position initiale de l'agent et le bord de la grille à $d = 8$.

La position de départ des agents est représentée, pour une grille de taille $m \times k$, avec les coordonnées X et Y : (Agent 1 (X,Y)) avec $0 \leq X < m$ et $0 \leq Y < k$.

4.1.2 Les résultats

Pour chaque test, on présente la moyenne de rondes pour atteindre le bord de la grille, la moyenne de rondes pour enregistrer la carte du bord de la grille, la moyenne de rondes pour se rencontrer sur le bord de la grille et le nombre total des rondes du rendez-vous. Pour une configuration de départ donnée, le nombre total de rondes du rendez-vous doit être le même pour les deux agents.

De façon générale les résultats démontrent que :

- Les deux agents font presque le même nombre de pas pour enregistrer la carte du bord du graphe (Maximum de pas égale à $3 \times (2 \times dimension1 + 2 \times dimension2)$)
- La configuration de départ des agents affecte le nombre total de pas effectués avant d'atteindre le rendez-vous :
 - L'agent qui arrive le premier au bord, effectue plus de pas avant de rencontrer l'autre agent.
 - Il y a une forte corrélation entre le nombre de pas effectués pour arriver au bord de la grille et le nombre de pas pour réaliser la rencontre sur le bord de la grille. L'agent qui arrive le premier au bord de la grille effectue plus de pas avant de réaliser la rencontre et le contraire est vrai pour l'agent qui arrive plus tard au bord de la grille.
- il y a une forte corrélation entre le nombre de pas pour réaliser la rencontre sur le bord de la grille et la taille de la grille. Plus la taille est grande plus le nombre de pas pour réaliser la rencontre est élevé.

TABLE 4.1 – Résultats pour le rendez-vous au bord d'une grille de taille 5×10

		Grille de taille 5×10			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour se rencontrer sur le bord de la grille	Nombre de rondes total
Configuration 1	Agent 1(1;1)	1.58	35.27	24.68	61.53
	Agent 2(2;1)	2.53	40	19	
Configuration 2	Agent 1(1;1)	1.77	35.27	34.31	71.35
	Agent 2(2;2)	10.68	36.15	24.52	
Configuration 3	Agent 1(1;1)	1.72	33.78	35.89	71.39
	Agent 2(3;3)	2.25	40.00	29.14	
Configuration 4	Agent 1(2;2)	11.12	36.48	7.8	55.4
	Agent 2(2;2)	11.12	36.48	7.8	
Configuration 5	Agent 1(2;2)	10.98	36.37	66.01	113.36
	Agent 2(3;3)	2.34	40	71.02	

TABLE 4.2 – Résultats pour le rendez-vous au bord d'une grille de taille 5×20

		Grille de taille 5×20			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour se rencontrer sur le bord de la grille	Nombre de rondes total
Configuration 1	Agent 1(1;1)	1.67	58.87	37	97.54
	Agent 2(2;1)	2.42	70	25.12	
Configuration 2	Agent 1(1;1)	1.64	60.76	68.14	130.54
	Agent 2(2;2)	11.14	62.44	56.96	
Configuration 3	Agent 1(1;1)	1.82	58.45	107.18	167.45
	Agent 2(3;18)	1.65	60.34	105.46	
Configuration 4	Agent 1(2;7)	14.16	58.66	57	129.82
	Agent 2(2;13)	16.02	60.55	35.25	
Configuration 5	Agent 1(2;7)	14.36	60.13	100.46	174.95
	Agent 2(3;18)	1.85	58.45	114.65	

TABLE 4.3 – Résultats pour le rendez-vous au bord d'une grille de taille 5×50

		Grille de taille 5×50			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour se rencontrer sur le bord de la grille	Nombre de rondes total
Configuration 1	Agent 1 (1;1)	1.77	137.05	178.47	317.29
	Agent 2 (2;1)	2.68	160	154.61	
Configuration 2	Agent 1 (1;1)	1.67	137.05	67.66	206.38
	Agent 2 (2;7)	13.84	135.52	57.02	
Configuration 3	Agent 1 (1;1)	1.59	128.89	253.56	384.04
	Agent 2 (3;48)	1.58	131.95	250.51	
Configuration 4	Agent 1 (2;7)	15.14	133.48	133.04	281.66
	Agent 2 (2;43)	13.74	135.01	132.91	
Configuration 5	Agent 1 (2;7)	14.44	135.01	133.33	282.78
	Agent 2 (3;48)	1.68	131.95	149.15	

TABLE 4.4 – Résultats pour le rendez-vous au bord d'une grille de taille 5×100

		Grille de taille 5×100			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour se rencontrer sur le bord de la grille	Nombre de rondes total
Configuration 1	Agent 1 (1;1)	1.65	251.42	368.71	621.78
	Agent 2 (2;1)	2.44	310	309.34	
Configuration 2	Agent 1 (1;1)	1.66	258.49	103.15	363.3
	Agent 2 (2;7)	14.28	257.48	91.54	
Configuration 3	Agent 1 (1;1)	1.67	264.55	327.71	593.93
	Agent 2 (3;97)	2.5	310	281.43	
Configuration 4	Agent 1 (2;7)	14.94	250.41	318.11	583.46
	Agent 2 (2;93)	13.52	264.55	305.93	
Configuration 5	Agent 1 (2;7)	15.26	264.55	365.58	645.39
	Agent 2 (3;97)	2.5	310	332.89	

TABLE 4.5 – Résultats pour le rendez-vous au bord d'une grille de taille 10×10

		Grille de taille 10×10			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour se rencontrer sur le bord de la grille	Nombre de rondes total
Configuration 1	Agent 1 (1;1)	1.76	52	172.04	225.8
	Agent 2 (2;1)	2.54	60	163.26	
Configuration 2	Agent 1 (1;1)	1.68	50.72	112.99	165.39
	Agent 2 (2;4)	20.08	44	101.31	
Configuration 3	Agent 1 (1;1)	1.62	51.84	46.81	100.27
	Agent 2 (8;8)	1.64	51.68	46.95	
Configuration 4	Agent 1 (4;4)	531.6	53.12	60.18	644.9
	Agent 2 (4;4)	498.6	54	92.3	
Configuration 5	Agent 1 (4;4)	512.72	52.32	385.02	950.06
	Agent 2 (8;8)	1.68	51.68	896.7	

TABLE 4.6 – Résultats pour le rendez-vous au bord d'une grille de taille 20×20

		Grille de taille 20×20			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour se rencontrer sur le bord de la grille	Nombre de rondes total
Configuration 1	Agent 1 (1;1)	1.74	112	30.41	144.15
	Agent 2 (2;1)	2.58	130	11.57	
Configuration 2	Agent 1 (1;1)	1.66	110.92	63.07	175.65
	Agent 2 (2;7)	21.8	94	59.85	
Configuration 3	Agent 1 (1;1)	1.77	114.16	81.89	197.82
	Agent 2 (18;18)	1.61	111.64	84.57	
Configuration 4	Agent 1 (10;7)	87129.23	130	29250.97	116510.2
	Agent 2 (10;13)	17893.56	94	98522.64	
Configuration 5	Agent 1 (10;7)	85994.74	130	42424.15	128548.89
	Agent 2 (18;18)	1.57	298.72	128248.6	

TABLE 4.7 – Résultats pour le rendez-vous au bord d'une grille de taille 50×50

		Grille de taille 50×50			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour se rencontrer sur le bord de la grille	Nombre de rondes total
Configuration 1	Agent 1 (1;1)	1.74	292	623.87	917.61
	Agent 2 (2;1)	2.6	340	575.01	
Configuration 2	Agent 1 (1;1)	1.66	293.92	126.83	422.41
	Agent 2 (2;7)	20.72	244	157.69	
Configuration 3	Agent 1 (1;1)	1.66	292	208.219	501.87
	Agent 2 (48;48)	1.64	291.04	209.19	
Configuration 4	Agent 1 (25;7)	89143.06	340	965660.01	1055143.07
	Agent 2 (25;43)	17830.86	244	1037068.21	
Configuration 5	Agent 1 (25;7)	88750.01	340	50341.23	139431.24
	Agent 2 (48;48)	1.63	298.72	139130.89	

TABLE 4.8 – Résultats pour le rendez-vous au bord d'une grille de taille 100×100

		Grille de taille 100×100			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour se rencontrer sur le bord de la grille	Nombre de rondes total
Configuration 1	Agent 1 (1;1)	1.48	593.96	2192.75	2788.19
	Agent 2 (2;1)	2.41	690	2095.78	
Configuration 2	Agent 1 (1;1)	1.76	601.8	221.63	825.19
	Agent 2 (2;7)	19	494	312.19	
Configuration 3	Agent 1 (1;1)	1.53	592	1580	2173.53
	Agent 2 (97;97)	14.76	590.04	1568.37	
Configuration 4	Agent 1	-	-	-	-
	Agent 2	-	-	-	
Configuration 5	Agent 1 (50;7)	90442.96	690	46020.1	137153.06
	Agent 2 (97;97)	14.14	584.16	136554.76	

4.2 Résultats pour la simulation du rendez-vous au centre d'une grille de taille $m \times k$, avec $2 < m \leq k$

4.2.1 Données d'entrée

Pour la Rencontre au centre de la grille on considère sept tailles de grilles différentes ayant toutes des dimensions impaires :

- 5×11
- 5×21
- 5×51
- 5×101
- 11×11
- 21×21
- 51×51

Pour chaque taille on considère les cinq configurations listées dans la figure 4.1. On s'attend à ce que la position du départ des agents affecte uniquement le nombre de pas pour atteindre le bord de la grille.

Pour chaque taille de grille et pour chaque configuration de départ, l'algorithme est exécuté 100 fois. Pour chaque exécution, le graphe est réinitialisé en redistribuant de façon aléatoire les numéros de ports sur les nœuds de la grille.

La distance maximale entre la position initiale de l'agent et le bord de la grille est $d = 8$. La position du départ des agents est représentée, pour une grille de taille $m \times k$, avec les coordonnées X et Y : (Agent 1 (X, Y)) avec $0 \leq X < m$ et $0 \leq Y < k$.

4.2.2 Les résultats

Pour chaque test, on présente la moyenne du nombre de rondes pour atteindre le bord de la grille, la moyenne du nombre de rondes pour faire le tour de la grille, la moyenne du nombre de rondes pour trouver le centre de la grille et la moyenne du nombre total des rondes du rendez-vous qui est égale au nombre total de rondes du dernier agent à trouver le centre de la grille.

De façon générale les résultats démontrent que :

- Les positions initiales des agents affectent seulement le nombre de pas effectués pour atteindre le bord de la grille

- Pour une grille de taille donnée, une fois l’agent sur le bord de la grille, il fait presque le même nombre de pas pour se rencontrer au centre de la grille peu importe sa position de départ initiale.
- Plus la position du départ de l’agent s’éloigne du bord de la grille, plus le nombre de pas effectués pour atteindre le bord de la grille l’emporte sur le nombre de pas effectués pour réaliser la rencontre (voir résultats pour les grilles de tailles 21×21 et 51×51)

TABLE 4.9 – Résultats pour le rendez-vous au centre d’une grille de taille 5×11

		Grille de taille 5×11			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour trouver le centre de la grille	Nombre de rondes total du rendez-vous
Configuration 1	Agent 1 (1;1)	1.68	36.07	181.95	228.92
	Agent 2 (2;1)	11.3	38.92	178.7	
Configuration 2	Agent 1 (1;1)	1.59	38.38	181.83	227.33
	Agent 2 (2;2)	2.4	43.00	181.94	
Configuration 3	Agent 1 (1;1)	1.63	37.63	182.74	221.99
	Agent 2 (3;3)	2.64	43.00	172.39	
Configuration 4	Agent 1 (2;2)	11.08	38.5	176.81	226.40
	Agent 2 (2;2)	11.08	38.5	176.81	
Configuration 5	Agent 1 (2;2)	2.56	43.00	171.15	227.15
	Agent 2 (3;3)	11.02	38.38	177.75	

TABLE 4.10 – Résultats pour le rendez-vous au centre d'une grille de taille 5×21

		Grille de taille 5×21			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour trouver le centre de la grille	Nombre de rondes total du rendez-vous
Configuration 1	Agent 1 (1;1)	1.66	63.80	387.96	463.72
	Agent 2 (2;1)	2.42	73.00	388.31	
Configuration 2	Agent 1 (1;1)	1.68	61.55	386.96	460.37
	Agent 2 (2;2)	11.51	65.99	383.27	
Configuration 3	Agent 1 (1;1)	1.72	59.25	386.13	447.09
	Agent 2 (3;18)	2.43	73.00	332.35	
Configuration 4	Agent 1 (2;7)	14.60	60.36	365.14	440.10
	Agent 2 (2;13)	14.04	60.60	347.22	
Configuration 5	Agent 1 (2;7)	14.00	60.12	365.46	439.59
	Agent 2 (3;18)	2.43	73.00	332.39	

TABLE 4.11 – Résultats pour le rendez-vous au centre d'une grille de taille 5×51

		Grille de taille 5×51			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour trouver le centre de la grille	Nombre de rondes total du rendez-vous
Configuration 1	Agent 1 (1;1)	2.41	163.00	1156.41	1441.15
	Agent 2 (2;1)	1.71	173.79	1301.66	
Configuration 2	Agent 1 (1;1)	1.73	135.22	1301.12	1437.98
	Agent 2 (2;7)	14.27	143.7	1279.54	
Configuration 3	Agent 1 (1;1)	1.71	137.79	1301.66	1441.15
	Agent 2 (3;48)	2.41	163.00	1156.41	
Configuration 4	Agent 1 (2;7)	13.89	136.18	1273.02	1423.09
	Agent 2 (2;43)	14.63	135.08	1160.34	
Configuration 5	Agent 1 (2;7)	13.37	137.26	1279.23	1429.87
	Agent 2 (3;48)	2.44	163.00	1156.23	

TABLE 4.12 – Résultats pour le rendez-vous au centre d'une grille de taille 5×101

		Grille de taille 5×101			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour trouver le centre de la grille	Nombre de rondes total du rendez-vous
Configuration 1	Agent 1 (1;1)	1.73	260.94	3826.25	4142.91
	Agent 2 (2;1)	2.66	313.00	3827.25	
Configuration 2	Agent 1 (1;1)	1.58	259.37	3826.03	4091.48
	Agent 2 (2;7)	13.15	274.09	3804.24	
Configuration 3	Agent 1 (1;1)	1.64	260.42	3826.11	4088.18
	Agent 2 (3;97)	2.71	313.00	3534.28	
Configuration 4	Agent 1 (2;7)	15.46	255.16	3804.13	4074.76
	Agent 2 (2;93)	14.41	260.42	3546.13	
Configuration 5	Agent 1 (2;7)	14.43	265.12	3804.30	4083.85
	Agent 2 (3;97)	2.59	313.00	3534.30	

TABLE 4.13 – Résultats pour le rendez-vous au centre d'une grille de taille 11×11

		Grille de taille 11×11			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour trouver le centre de la grille	Nombre de rondes total du rendez-vous
Configuration 1	Agent 1 (1;1)	1.74	58.10	557.95	660.91
	Agent 2 (2;1)	2.44	67.00	591.47	
Configuration 2	Agent 1 (1;1)	1.68	56.77	555.00	613.45
	Agent 2 (2;4)	20.02	49.00	515.84	
Configuration 3	Agent 1 (1;1)	1.72	59.34	564.24	625.31
	Agent 2 (8;8)	14.15	56.66	541.91	
Configuration 4	Agent 1 (4;4)	549.77	57.91	551.05	1158.73
	Agent 2 (4;4)	549.77	57.91	551.05	
Configuration 5	Agent 1 (4;4)	534.11	58.47	552.91	1145.48
	Agent 2 (8;8)	15.87	59.80	516.61	

TABLE 4.14 – Résultats pour le rendez-vous au centre d'une grille de taille 21×21

		Grille de taille 21×21			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour trouver le centre de la grille	Nombre de rondes total du rendez-vous
Configuration 1	Agent 1 (1;1)	1.68	113.89	3226.05	3709.89
	Agent 2 (2;1)	2.58	137.00	3570.31	
Configuration 2	Agent 1 (1;1)	1.80	120.55	3320.73	3443.08
	Agent 2 (2;7)	21.59	99.00	3051.19	
Configuration 3	Agent 1 (1;1)	1.62	115.67	3277.67	3394.97
	Agent 2 (18;18)	14.43	119.16	3219.71	
Configuration 4	Agent 1 (10;7)	88338.27	137.91	3554.34	92849.37
	Agent 2 (10;13)	89196.03	99.00	3554.34	
Configuration 5	Agent 1 (10;7)	87446.11	137.47	3557.91	91140.64
	Agent 2 (18;18)	13.63	115.93	3273.55	

TABLE 4.15 – Résultats pour le rendez-vous au centre d'une grille de taille 51×51

		Grille de taille 51×51			
		Nombre de rondes pour arriver au bord de la grille	Nombre de rondes pour faire le tour de la grille	Nombre de rondes pour trouver le centre de la grille	Nombre de rondes total du rendez-vous
Configuration 1	Agent 1 (1;1)	1.70	298.54	44614.14	49649.14
	Agent 2 (2;1)	2.70	347.00	49299.44	
Configuration 2	Agent 1 (1;1)	1.82	298.00	44220.18	44520.00
	Agent 2 (2;7)	20.67	249.00	41434.95	
Configuration 3	Agent 1 (1;1)	15.21	294.87	44102.82	45765.68
	Agent 2 (48;48)	1.66	299.04	45464.98	
Configuration 4	Agent 1 (25;7)	91290.88	347.00	49229.22	141545.24
	Agent 2 (25;43)	92067.02	249.00	49229.22	
Configuration 5	Agent 1 (25;7)	82946.33	347.00	49228.59	132521.92
	Agent 2 (48;48)	15.31	303.80	43891.28	

4.3 Résultats pour la simulation du rendez-vous au centre d'une grille de taille $2 \times k$

4.3.1 Données d'entrée

Vu la complexité (exigence en ressource) élevée de l'algorithme `Rencontre au centre de la grille de taille $2 \times k$` on considère (pour des fins de démonstration) trois tailles de grilles relativement petites :

- 2×5
- 2×11
- 2×15

Pour chaque taille on considère les trois configurations de départ suivantes :

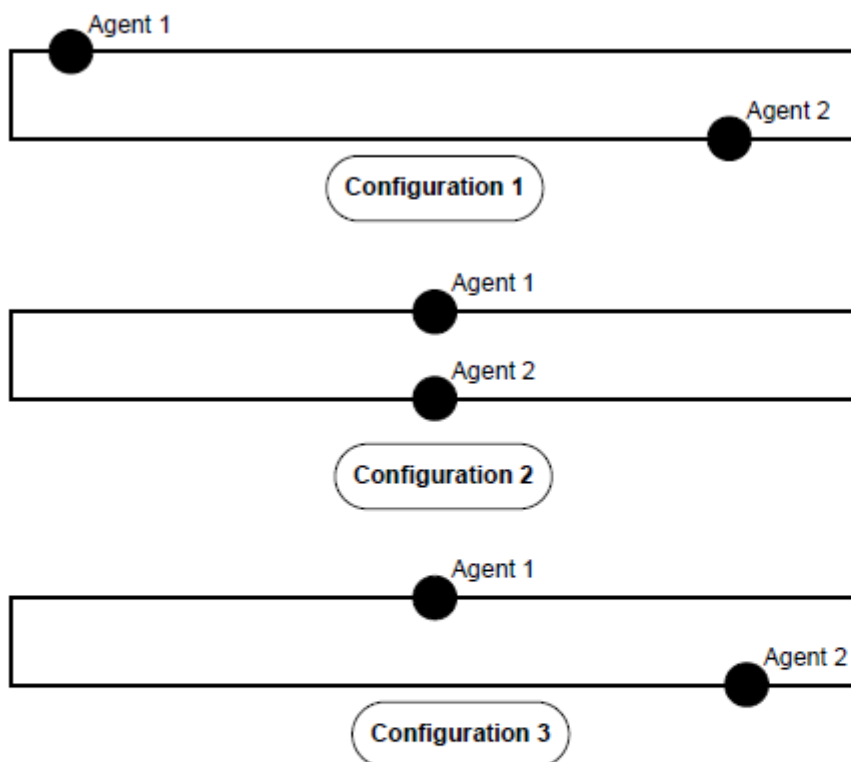


FIGURE 4.2 – Les trois configurations de départ pour le rendez-vous au centre de l'échelle

Pour les trois configurations on s'attend à ce que les deux agents fassent autant de pas pour trouver le centre de la grille.

Pour chaque taille de grille et pour chaque configuration de départ, l'algorithme est exécuté 100 fois. Pour chaque exécution, le graphe est réinitialisé en redistribuant de façon aléatoire les numéros de ports sur les nœuds de la grille.

La position du départ des agents est représentée, pour une grille de taille $2 \times k$, avec les coordonnées X et Y : (Agent 1 (X,Y)) avec $0 \leq X \leq 1$ et $0 \leq Y < k$.

4.3.2 Les résultats

Pour chaque test, on présente la moyenne du nombre de rondes pour trouver le centre de la grille, la moyenne du nombre de rondes pour réaliser la rencontre au centre de l'échelle et la moyenne du nombre total de rondes.

De façon générale les résultats démontrent que :

- Les deux agents effectuent presque le même nombre de pas pour trouver le centre de la grille.
- Une petite augmentation de la taille de l'échelle fait monter de façon exponentielle le nombre de pas effectués pour trouver le centre de la grille.

TABLE 4.16 – Résultats pour le rendez-vous au centre de l'échelle de taille 2×5

		Grille de taille 2×5		
		Nombre de rondes pour trouver le centre de la grille	Nombre de rondes pour se rencontrer au centre de la grille	Nombre de rondes total
Configuration 1	Agent 1 (0;1)	253	17.4	270.4
	Agent 2 (1;3)	253	17.4	
Configuration 2	Agent 1 (0;2)	264.44	5.23	269.67
	Agent 2 (1;2)	253	16.67	
Configuration 3	Agent 1 (0;2)	263.64	8.86	272.5
	Agent 2 (1;3)	253	19.5	

TABLE 4.17 – Résultats pour le rendez-vous au centre de l'échelle de taille 2×11

		Grille de taille 2×11		
		Nombre de rondes pour trouver le centre de la grille	Nombre de rondes pour se rencontrer au centre de la grille	Nombre de rondes total
Configuration 1	Agent 1 (0;1)	103000	25.6	103025.6
	Agent 2 (1;8)	103000	25.6	
Configuration 2	Agent 1 (0;5)	104565.2	1134.8	105700
	Agent 2 (1;5)	103000	2700	
Configuration 3	Agent 1 (0;5)	104264.2	305.03	104569.23
	Agent 2 (1;8)	103000	1569.23	

TABLE 4.18 – Résultats pour le rendez-vous au centre de l'échelle de taille 2×15

		Grille de taille 2×15		
		Nombre de rondes pour trouver le centre de la grille	Nombre de rondes pour se rencontrer au centre de la grille	Nombre de rondes total
Configuration 1	Agent 1 (0;1)	5437370	13	5437383
	Agent 2 (1;14)	5437370	13	
Configuration 2	Agent 1 (0;7)	5437360	11.2	5437371.2
	Agent 2 (1;7)	5437358.4	12.8	
Configuration 3	Agent 1 (0;7)	5437374	1324	5438698
	Agent 2 (1;14)	5437374	1324	

4.4 Résultats pour la simulation du rendez-vous aléatoire avec et sans retour

4.4.1 Données d'entrée

Pour les algorithmes aléatoires on considère les grilles de tailles

- 5×5
- 5×10
- 5×20
- 5×50
- 5×100
- 10×10
- 20×20
- 50×50

Pour chaque taille de grille on considère les configurations tel qu' illustré dans la figure 4.1.

Pour chaque taille de grille et pour chaque configuration de départ, l'algorithme est exécuté 100 fois. Pour chaque exécution, le graphe est réinitialisé en redistribuant de façon aléatoire les numéros de ports sur les nœuds de la grille.

À chaque ronde, la fonction aléatoire utilisée génère un nombre aléatoire k avec $0 \leq k \leq d$ avec d le degré du nœud courant. Dans le cas où $k = d$, l'agent reste sur le nœud courant, sinon il visite le nœud voisin à travers le port numéro k .

Dans le cas du rendez-vous aléatoire sans retour, la fonction aléatoire exclue le numéro

de port par lequel l'agent a visité le nœud courant.

La position du départ des agents est représentée, pour une grille de taille $m \times k$, avec les coordonnées X et Y : (Agent 1 (X,Y)) avec $0 \leq X < m$ et $0 \leq Y < k$.

4.4.2 Les résultats

De façon générale on constate que :

- Plus la taille de la grille est grande plus le nombre de rondes pour réaliser le rendez-vous est élevé.
- L'algorithme aléatoire sans retour est plus efficace que celui avec retour.
- Les algorithmes aléatoires sont plus efficaces que les algorithmes déterministes.

4.4.3 Résultats pour la simulation du rendez-vous aléatoire avec retour

TABLE 4.19 – Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 5×5

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	31.62
	Agent2 (1;1)	
Configuration 2	Agent1 (2;2)	31.72
	Agent2 (3;3)	
Configuration 3	Agent1 (1;1)	39.5
	Agent2 (4;4)	
Configuration 4	Agent1 (2;2)	29.4
	Agent2 (2;2)	
Configuration 5	Agent1 (3;3)	41.56
	Agent2 (1;1)	

TABLE 4.20 – Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 5×10

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	42.7
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	73.8
	Agent2 (2;5)	
Configuration 3	Agent1 (1;1)	48.65
	Agent2 (8;8)	
Configuration 4	Agent1 (2;5)	61.69
	Agent2 (2;5)	
Configuration 5	Agent1 (2;5)	82.52
	Agent2 (8;8)	

TABLE 4.21 – Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 5×20

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	58.91
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	204.04
	Agent2 (2;10)	
Configuration 3	Agent1 (1;1)	394.64
	Agent2 (4;19)	
Configuration 4	Agent1 (2;10)	293.32
	Agent2 (2;10)	
Configuration 5	Agent1 (2;10)	292.79
	Agent2 (4;19)	

TABLE 4.22 – Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 5×50

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	170.18
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	1961.68
	Agent2 (2;25)	
Configuration 3	Agent1 (1;1)	1401.84
	Agent2 (48;48)	
Configuration 4	Agent1 (2;25)	1354.3
	Agent2 (2;43)	
Configuration 5	Agent1 (2;25)	1199.98
	Agent2 (48;48)	

TABLE 4.23 – Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 5×100

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	4879.62
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	7434.34
	Agent2 (2;50)	
Configuration 3	Agent1 (1;1)	4151.23
	Agent2 (48;48)	
Configuration 4	Agent1 (2;50)	4740.1
	Agent2 (2;48)	
Configuration 5	Agent1 (2;50)	5086.48
	Agent2 (98;98)	

TABLE 4.24 – Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 10×10

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	169.95
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	115.03
	Agent2 (2;5)	
Configuration 3	Agent1 (1;1)	143.6
	Agent2 (8;8)	
Configuration 4	Agent1 (4;4)	89.56
	Agent2 (4;4)	
Configuration 5	Agent1 (4;4)	231
	Agent2 (4;8)	

TABLE 4.25 – Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 20×20

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	446.52
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	772.45
	Agent2 (2;10)	
Configuration 3	Agent1 (1;1)	1000.68
	Agent2 (18;18)	
Configuration 4	Agent1 (10;10)	675
	Agent2 (10;13)	
Configuration 5	Agent1 (10;10)	812.4
	Agent2 (18;18)	

TABLE 4.26 – Résultats pour le rendez-vous aléatoire avec retour dans une grille de taille 50×50

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	1144.66
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	6112.21
	Agent2 (2;25)	
Configuration 3	Agent1 (1;1)	7057.38
	Agent2 (48;48)	
Configuration 4	Agent1 (25;25)	5803.3
	Agent2 (25;28)	
Configuration 5	Agent1 (25;25)	8576.34
	Agent2 (48;48)	

4.4.4 Résultats pour la simulation du rendez-vous aléatoire sans retour

TABLE 4.27 – Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 5×5

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	27.58
	Agent2 (1;1)	
Configuration 2	Agent1 (2;2)	24.2
	Agent2 (3;3)	
Configuration 3	Agent1 (1;1)	31.61
	Agent2 (4;4)	
Configuration 4	Agent1 (2;2)	29.52
	Agent2 (2;2)	
Configuration 5	Agent1 (3;3)	33.12
	Agent2 (1;1)	

TABLE 4.28 – Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 5×10

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	42.61
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	63.97
	Agent2 (2;5)	
Configuration 3	Agent1 (1;1)	66.55
	Agent2 (8;8)	
Configuration 4	Agent1 (2;5)	38.8
	Agent2 (2;5)	
Configuration 5	Agent1 (2;5)	67
	Agent2 (8;8)	

TABLE 4.29 – Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 5×20

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	85.05
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	140.34
	Agent2 (2;10)	
Configuration 3	Agent1 (1;1)	188.48
	Agent2 (4;19)	
Configuration 4	Agent1 (2;10)	188.76
	Agent2 (2;10)	
Configuration 5	Agent1 (2;10)	231.92
	Agent2 (4;19)	

TABLE 4.30 – Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 5×50

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	107.18
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	917.72
	Agent2 (2;25)	
Configuration 3	Agent1 (1;1)	883.27
	Agent2 (48;48)	
Configuration 4	Agent1 (2;25)	827.45
	Agent2 (2;43)	
Configuration 5	Agent1 (2;25)	1231.12
	Agent2 (48;48)	

TABLE 4.31 – Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 5×100

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	105.4
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	2618.86
	Agent2 (2;50)	
Configuration 3	Agent1 (1;1)	3239.18
	Agent2 (48;48)	
Configuration 4	Agent1 (2;50)	2542.42
	Agent2 (2;48)	
Configuration 5	Agent1 (2;50)	4907.5
	Agent2 (98;98)	

TABLE 4.32 – Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 10×10

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	87.4
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	132.5
	Agent2 (2;5)	
Configuration 3	Agent1 (1;1)	158.2
	Agent2 (8;8)	
Configuration 4	Agent1 (4;4)	110.74
	Agent2 (4;4)	
Configuration 5	Agent1 (4;4)	175.11
	Agent2 (4;8)	

TABLE 4.33 – Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 20×20

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	353.2
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	722.4
	Agent2 (2;10)	
Configuration 3	Agent1 (1;1)	854.33
	Agent2 (18;18)	
Configuration 4	Agent1 (10;10)	536.14
	Agent2 (10;13)	
Configuration 5	Agent1 (10;10)	794.2
	Agent2 (18;18)	

TABLE 4.34 – Résultats pour le rendez-vous aléatoire sans retour dans une grille de taille 50×50

		Nombre de rondes pour se rencontrer
Configuration 1	Agent1 (2;1)	1824.33
	Agent2 (1;1)	
Configuration 2	Agent1 (1;1)	6012.33
	Agent2 (2;25)	
Configuration 3	Agent1 (1;1)	5703.75
	Agent2 (48;48)	
Configuration 4	Agent1 (25;25)	4124.25
	Agent2 (25;28)	
Configuration 5	Agent1 (25;25)	5729.35
	Agent2 (48;48)	

Chapitre 5

Conclusion

Grâce aux simulations, on a pu constater que les algorithmes aléatoires sont plus performants que les algorithmes déterministes pour le problème du rendez-vous dans les grilles.

La procédure `Recherche exponentielle du bord de la grille` est la plus coûteuse de toutes les procédures que nous avons utilisées.

Comme mentionné dans le chapitre 3, il existe une méthode polynomiale pour explorer la grille, basée sur [20], mais elle est beaucoup plus compliquée et pas efficace non plus. Le polynôme est de haut degré. Selon nos connaissances, aucune méthode efficace d'explorer une grille ou même d'atteindre le bord sans marquer les nœuds n'est connue à présent. Afin de palier à cette lacune en performance on a présenté un algorithme dans l'article [19] qui utilise un jeton stationnaire pour trouver le bord d'une grille de taille $m \times k$ avec $2 < m < k$ en un temps de $O(m^2)$.

Une amélioration importante des résultats de ce mémoire serait de trouver un algorithme de recherche du bord d'une grille de taille $m \times k$ avec $2 < m < k$ travaillant en temps $O(m)$ ou $O(m^2)$ si un tel algorithme existe. De même, il serait intéressant de trouver l'algorithme optimal pour explorer une échelle $2 \times k$.

Bibliographie

- [1] Alpern, S., and Gal, S. The theory of search games and rendezvous. Kluwer Academic Publishers, 2003.
- [2] Bampas, E., Czyzowicz, J., Gasieniec, L., Ilcinkas, D., and Labourel, A. Almost optimal asynchronous rendezvous infinite multidimensional grids. Proc. of 24th International Conference on Distributed Computing, DISC 2010, pp. 297-311.
- [3] Baumann, J., Hohl, F., Rothermel, K., Strasser, M. : Mole : Concepts of a Mobile Agent System. Journal World Wide Web, Volume 1, Issue 3, 1998, pp. 123-137.
- [4] Carzaniga, A., Picco, G., Vigna, G. : Designing Distributed Applications with Mobile Code Paradigm. Proc. of 19th International Conference on Software Engineering, ICSE 1997, pp.22-32 Proc.
- [5] Chia, T., Kannapan, S. : Strategically Mobile Agents. Proc. of 1st International Workshop on Mobile Agents, MA 1997, pp. 149-161.
- [6] Czyzowicz, J., Kosowski, A., and Pelc, A. How to meet when you forget : log-space rendezvous in arbitrary graphs. Proc. of 29th ACM SIGACT SIGOPS Symposium on Principles of Distributed Computing, PODC 2010, pp. 450-459
- [7] Czyzowicz, J., Labourel, A., and Pelc, A. How to meet asynchronously (almost) everywhere. Proc. of 21st Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, pp. 22-30.
- [8] Dessmark, A., Fraigniaud, P., Kowalski, D. R., and Pelc, A. Deterministic rendezvous in graphs. Algorithmica, Volume 46, Issue 1, 2006 , pp. 69-96.
- [9] Elouasbi, S., Pelc, A. : Time of Anonymous Rendezvous in Trees : Determinism vs. Randomization. Proc. of 19th International Colloquium on Structural Information and Communication Complexity, SIROCCO 2012, pp. 291-302.
- [10] Flocchini, P., Kranakis, E., Krizanc, D., Luccio, F., Santoro, N., and Sawchuk, C. Mobile agents rendezvous when tokens fail. Proc. of 11th International Colloquium on Structural Information and Communication Complexity, SIROCCO 2004, pp. 161-172.
- [11] Guilbault, S., Pelc, A. : Asynchronous Rendezvous of Anonymous Agents in Arbitrary Graphs. Proc. of 15th International Conference on Principles of Distributed Systems, OPODIS 2011, pp. 421-434.

- [12] Kawai, S., Ooshita, F., Kakugawa, H., Masuzawa, T. : Randomized Rendezvous of Mobile Agents in Anonymous Unidirectional Ring Networks. Proc. of 19th International Colloquium on Structural Information and Communication Complexity, SIROCCO 2012, pp. 303-314.
- [13] Kowalski, D., and Malinowski, A. How to meet in an anonymous network. Proc. of 13th International Colloquium on Structural Information and Communication Complexity, SIROCCO 2006, pp. 44-58.
- [14] Kranakis, E., and Krizanc, D. An algorithmic theory of mobile agents. Proc. of Second Symposium on Trustworthy Global Computing, TGC 2006, pp. 86-97.
- [15] Kranakis, E., Krizanc, D., and Markou, E. Mobile agent rendezvous in a synchronous torus. Proc. of 7th Latin American Symposium on Theoretical Informatics, LATIN 2006, pp. 653-664.
- [16] Kranakis, E., Krizanc, D., and Morin, P. Randomized rendez-vous with limited memory. Proc. of 8th Latin American Symposium on Theoretical Informatics, LATIN 2008, pp. 605-616.
- [17] Kranakis, E., Krizanc, D., Santoro, N., and Sawchuk, C. Mobile agent rendez-vous in a ring. Distributed Computing Systems, Proc. of 23rd IEEE International Conference on Distributed Computing Systems, ICDCS 2003, pp. 592.
- [18] Pelc, A., Deterministic rendezvous in networks : A Comprehensive Survey. Networks, Volume 59, Issue 3, 2012, pp. 331-347.
- [19] Pelc, A., Tiane, A. Efficient Grid Exploration with a Stationary Token, manuscript.
- [20] Reingold, O. Undirected Connectivity in Log-Space, Journal of the ACM 55 (2008)
- [21] Singelee, D., Preneel, B. : Secure E-commerce using Mobile Agents on Untrusted Hosts, Computer Security and Industrial Cryptography. Proc of COSIC, Internal Report, 2004.
- [22] Ta-Shma, A., and Zwick, U. Deterministic rendezvous, treasure hunts and strongly universal exploration sequences. Proc. of 18th Annual ACM-SIAM Symposium on Discrete algorithms, SODA 2007, pp. 599-608