

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université 08 Mai 1945-Guelma

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de fin d'étude de Master

Filière : Informatique

Spécialité : Ingénierie des Medias

Thème :

**Stratégie d'exploration multi agent pour les environnements
inconnus**

Sous la direction de :

Melle. ZEDADRA Ouarda

Présenté par :

Nedjoua.Brahmia

Rima .Boulahia

Juin 2013

Remerciements

Il est coutume d'avoir une page de remerciement dans tout mémoire que l'on écrit.... Mais qui vais-je onc remercier ?

Nous tenons tout d'abord à remercier dieu qui nous a donnés le pouvoir et la volonté pendant 17 ans consécutifs jusqu'à ce jour de fin d'études, tout en lui demandant de nous apporter davantage pour en continuer

Après réflexion nous avons trouvé quelques personnes qu'ils nous tenaient à cœur de remercier plus particulièrement.....

Nous tenons tout d'abord à remercier notre encadreuse Mlle .Zdadra Warda qui nous a encadré, et qui n'a épargné aucun effort pour nous orienter afin qu'on puisse mener à bien ce projet

Mes sincères remerciements aux membres de jury de nous faire l'honneur de juger cette humble thèse

Nous adressons notre profonde gratitude à l'ensemble du corps professoral et administratif du département d'informatique ayant contribué notre formation

Nos remerciements vont aussi nos familles et à nos amis et à tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire.

Dédicaces

*« Louange à don dieu pour sa clémence et sa bénédiction que nous avons accordé notre faveur par elle et au son nom nous avons commencé les chemins de la science que nous a
vous pris.. »*

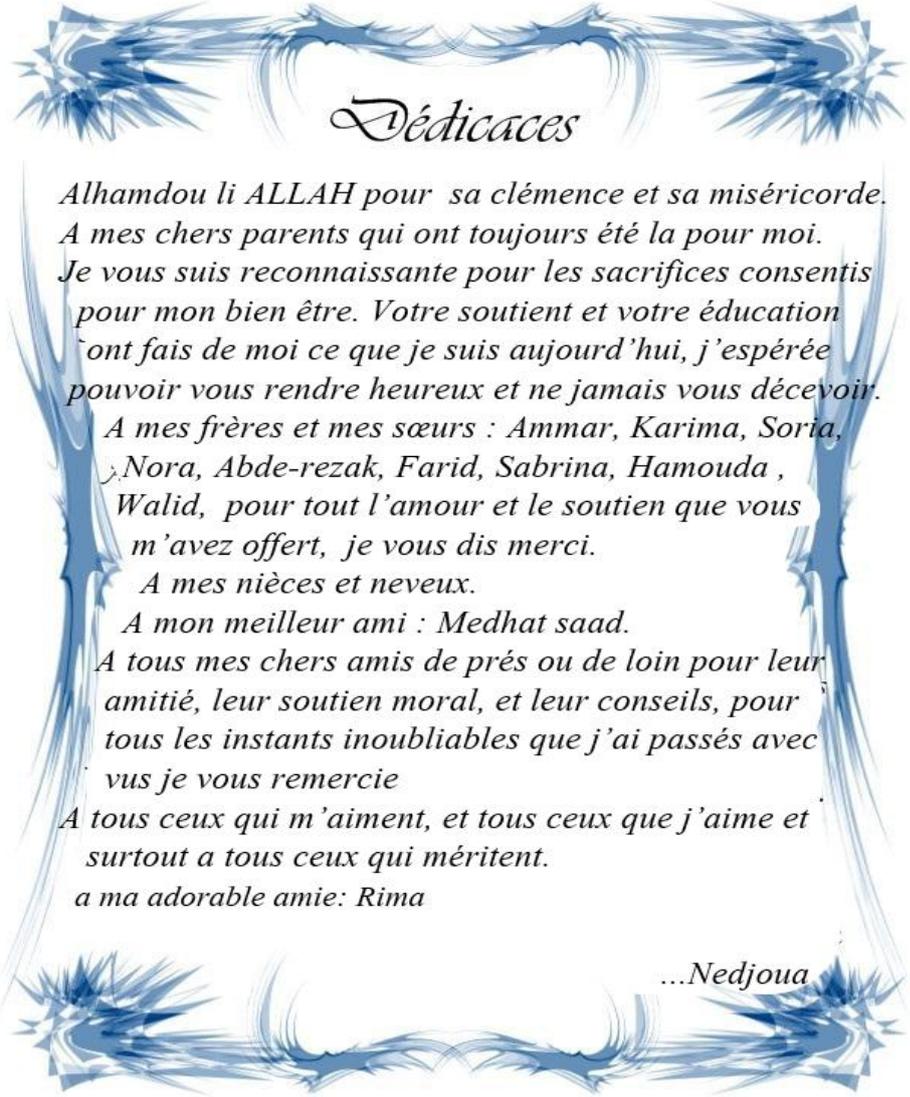
Au corps et au cœur de ma famille

- Mon cher père et ma tendre mère pour leur soutien avisé leur indulgence, fidele et exigeant du quel ce mémoire doit beaucoup.*
- A mes freres malek , haka, azzedin, allaoia*
- Mes chere sœurs badia et dalila pour leur soutien morale et leur sacrifices le long de ma formation*
- A toute ma famille a laquelle j'adresse mon profond respect et mon indéfectible attachement*
- A ma deuxième famille maman zohra, mes sœur merieme et kamila, mon frère bilel*
- A ma charmante amie selma*
- A ma merveilleuse binôme nedjoua*

- A mon mari Faouzi, sans qui rien n'est possible*

A tous ceux qui ont une relation de proche ou de loin avec la réalisation du présente rapport

...Rima



Dédicaces

Alhamdou li ALLAH pour sa clémence et sa miséricorde.

A mes chers parents qui ont toujours été la pour moi.

Je vous suis reconnaissante pour les sacrifices consentis pour mon bien être. Votre soutien et votre éducation ont fais de moi ce que je suis aujourd'hui, j'espérée pouvoir vous rendre heureux et ne jamais vous décevoir.

A mes frères et mes sœurs : Ammar, Karima, Soria,

*، Nora, Abde-rezak, Farid, Sabrina, Hamouda ,
Walid, pour tout l'amour et le soutien que vous m'avez offert, je vous dis merci.*

A mes nièces et neveux.

A mon meilleur ami : Medhat saad.

A tous mes chers amis de près ou de loin pour leur amitié, leur soutien moral, et leur conseils, pour tous les instants inoubliables que j'ai passés avec vous je vous remercie

A tous ceux qui m'aiment, et tous ceux que j'aime et surtout a tous ceux qui méritent.

a ma adorable amie: Rima

...Nedjoua

Résumé

La couverture d'un environnement inconnu par un ensemble de robots font partie des applications parmi les plus populaires en Intelligence Artificielle Distribuée .De nombreux modèles ont été développés pour trouver des règles simples permettant à des robots de couvrir leur environnement pour des tâches de couverture ou bien d'exploration et collecte. La métaphore la plus invoquée est celle des insectes, en grande partie parce ce que ces animaux ont développé des techniques collectives assez étonnantes. Nous adopterons dans ce mémoire l'approche réactive pour concevoir un comportement d'agrégation stratégique d'un groupe de robots autonomes .Le comportement des escargots pour la marche de chaque robot. Dans ce mémoire, nous étendons l'algorithme SMRSA de sorte à ce qu'il prend en compte une observabilité locale totale, des coupures de communication. Nous appliquons notre nouvelle SMRSA à une application sous JAVA consistant en la couverture multi-robot d'un environnement inconnu où chaque robot calcule localement une stratégie qui minimise les interactions entre les robots et maximise la couverture de l'espace par l'équipe même sous contraintes de pas communication entre les agents. Nous présentons des résultats expérimentaux issus de scénarios d'exécution.

Mots clés : intelligence artificiel distribué, système multi-agents, les agents réactifs, les agents, l'exploration des environnements inconnus, la couverture

Sommaire

Résumé	
Sommaire.....	1
Liste des figures.....	4
Introduction général	6
Chapitre I : état de l'art	
Introduction.....	9
I. Systèmes multi-agents: définition et principes	9
I.1. l'intelligence artificielle distribuée.....	10
I.2. Agents : concepts fondamentaux.....	10
I.2.1. Définition.....	10
I.2.2. Classification d'agent	11
I.2.3. Architecture d'agent	13
I.3. Principes des SMA.....	16
I.4. Interactions dans les SMA.....	17
I.5. Organisations dans les SMA	18
I.6. Communication dans les systèmes multi-agents	19
I.6.1. Définition par Russel	19
I.7. Coopération	20
I.8. Les mécanismes de coordination multi-agent	20
I.8.1. Structuration organisationnelle	22
I.8.2. Allocation	22
I.8.3. Planification	23
I.8.4.Négociation.....	24
I.8.5.Coordination réactive.....	24
I.9. Simulation multi agent	26

I.10. Avantage des SMA	27
I.11. Problèmes des SMA	28
I.12. Champs d'application des systèmes multi-agents	29
I.13. Conclusion.....	29

Chapitre II : couverture et exploration des environnements inconnus

Introduction	31
I. Les algorithmes de couverture	32
I.1. Les algorithmes de couverture pour un seul robot	33
. Algorithme Spanning Tree Couvrant (STC)	35
. Algorithme Spiral STC	39
I.2. Les algorithmes de couverture multi robots	40
. Algorithme MRSAM	42
. nouvel algorithme de couverture multi-robot (MCF)	45
I.3. Les champs de potentiels artificiels	46
I.4 Comparaison des résultats des algorithmes de couverture mono-robot et multi-robots	40
I.4.1. MFC & STC & MSTC	47
I.4.2. Off-line STC & On-line STC & Ant-like STC	47
II. L'exploration d'un environnement inconnu.....	48
II.1. Définition	48
II.2. Les algorithmes d'exploration	49
II.2.1. Frontière la plus proche.....	49
II.2.2. Glouton.....	51
II.3. Problème d'exploration.....	51
III. Conclusion	52

Chapitre III : Conception et réalisation

Introduction.....	53
I.Conception.....	53

I.1.Objectifs de l'application	53
I.2.Modélisation du problème.....	63
I.2.1.Modélisation de l'environnement.....	54
I.2.2.Modélisation des agents	54
I.3.Comportement des agents	55
II.Réalisation.....	56
II.1.Environment de programmation	57
1)Aspect matériel.....	57
2)Environnement de développement	57
II.2.Présentation de l'application.....	58
II.3.Résultats expérimentaux	62
IV. Conclusion	67
Conclusion Générale et perspectives	68
Bibliographie.....	70

Liste des figures

Chapitre 1: état de l'art

Figure 1.1: Axe pratique d'évaluation de la capacité d'un agent à accomplir individuellement des tâches complexes et à planifier ses actions	12
Figure 1.2: Classification d'agents	13
Figure 1.3: Architecture BDI	14
Figure 1.4: Architecture subsumption	15
Figure 1.5: Les composantes principales d'un SMA	16
Figure 1.6: Communication directe	19
Figure 1.7: Principaux mécanismes de coordination multi-agent.....	21
Figure 1.8: Exemples de structures organisationnelles.....	22
Figure 1.9: Résolution d'un problème par un SMA	23
Figure 1.10: Système de planification multi-agent (exemple)	24
Figure 1.11: Exemple de protocole de négociation entre deux agents	25
Figure 1.12: Principes de la simulation multi-agent	27

Chapitre 2 : les algorithmes de couverture et exploration des environnements inconnus

Figure 2.1 : Une couverture optimale ne signifie pas nécessairement une patrouille optimale	33
Figure 2.2: Un spanning tree (arcs bleus) d'un graphe grille	34

Figure 2.3:Un cycle hamiltonien réalisé à partir d'un spanning tree sur un environnement grille de 8x8 nœuds.....	35
Figure 2.4: Exemple de l'algorithme STC.....	36
Figure 2.5 : Un exemple d'exécution de l'algorithme STC off ligne	37
Figure 2.6: La fourmi peut se déplacer à carreaux ou des sommets, T1, T2, T3 et T4. (a)Le cercle est la plage du capteur efficace de la fourmi. (b) La forme équivalente dans un graphe non orienté	38
Figure 2.7: (a) une seule face, (b) à double face bord, (c) doublement de nœud à une cellule déconnectée	40
Figure 2.8: exemple de MSTC en fonctionnement	42
Figure 2.9: La région du disque recherché par SAD1 doublé à chaque étape	43
Figure 2.10: Un groupe de deux robots lancés par MRSAM à la recherche de la cible.....	44
Figure 2.11: exemple de l'algorithme MFC	46
Figure 2.12 : MFC et MSTC contre STC	47
Figure 2.13: Coordination implicite avec l'algorithme frontière la plus proche	50
Figure 2.14: Résultat de l'assignation avec l'algorithme Frontière la plus proche	50
Figure 2.15: Illustration de l'allocation de frontière avec l'algorithme glouton	51
 Chapitre III : Conception et réalisation	
Figure 3.1 : Interface principale	59
Figure 3.2 : Création d'environnement(a)	60
Figure 3.2 : Création d'environnement(b)	60
Figure 3.2 : Création d'environnement(c)	61
Figure 3.2 : Création d'environnement(d)	61
Figure 3.3: Lancer la simulation	62
Figure 3.4: Division de l'environnement selon le paramètre de rayon de carré	62

Figure 3.5: (etape1) lancer la couverture mono-robot	64
Figure 3.5: (Etape2) déplacement de robot vers la deuxième région	63
Figure 3.5: (Etape 3) déplacement de robot vers la troisième région	64
Figure 3.5: (Etape 4) couvrir la quatrième région	65
Figure 3.6 : (Eetape 1) lancer la couverture multi-robot	65
Figure 3.6 : (Eetape 2) lancer la couverture multi-robot	66

Introduction général:

L'intelligence est une notion difficile à cerner. Derrière ce terme se cachent de nombreuses acceptions différentes. Un système pourra être qualifié d'intelligent parce que pour un observateur extérieur, il semblera doté de capacités cognitives habituellement attribuées à l'homme, ou parce qu'il cherchera à reproduire les mécanismes par lesquels l'homme ou l'animal prend des décisions complexes.

L'intelligence artificielle est inspirée de la métaphore du penseur solitaire : les chercheurs dans ce domaine ont cherché à produire des programmes isolés, en émulant les processus cognitifs humains pour résoudre des problèmes complexes. Les problèmes sont parfois naturellement posés de manière distribuée ; on suppose qu'il existe un certain nombre d'entités capables d'agir et d'interagir ; ces problèmes sont ainsi inscrits dans une branche de l'IA : intelligence artificielle distribuée (IAD). L'apparition de l'intelligence artificielle distribuée (IAD) a remis en question l'étude des systèmes constitués d'un agent. Elle a pour objectif de réaliser des organisations de systèmes, capables de résoudre des problèmes par le biais d'un raisonnement symbolique. La métaphore du penseur solitaire a été remise en question et s'est accompagnée d'une nouvelle problématique : celle de l'interaction entre plusieurs entités. Une nouvelle question se pose alors : Comment un agent peut-il prendre en considération la présence d'autres acteurs dans le système, pour interagir au mieux avec eux ?

Les robots mobiles jouent un rôle important dans de nombreux domaines, parmi lesquels : la police militaire et applications (par exemple, le déminage, les véhicules de patrouille armés, la surveillance, la bombe désarmement), l'industrie (par exemple, transport et de manutention des matériaux), l'espace (par exemple, le plan-exploration monétaire, et l'acquisition des matières sur des planètes lointaines).

Néanmoins, l'intérêt indéniable de la robotique mobile est d'avoir permis d'augmenter considérablement nos connaissances sur la localisation et la navigation de systèmes autonomes. La gamme des problèmes potentiellement soulevés par le plus simple des robots mobiles à roues.

La planification du mouvement du robot s'agit de synthétiser un chemin et une trajectoire dans l'espace des configurations du robot, c'est une tâche fondamentale dans les opérations robotiques mobiles, La trajectoire est habituellement un ensemble de commandes de sortie, comme direction de déplacement, la distance et la vitesse, sur la base de variables d'entrée tels que la position actuelle.

La robotique est surtout les robots fourrageurs sont des applications possible des systèmes multi agents, elle est caractérisé par un ensemble d'agents (hétérogènes ou homogènes), qui doivent coopérer et coordonner leurs actions pour parvenir à réaliser leur objectif commun qui est la recherche et le collecte d'un échantillon de minerai précis dans un environnement inconnu, contraint ou non, à une base bien spécifique (dépôt). Ce problème est un problème benchmark dans le domaine des SMA, parce que il inclut dedans un ensemble de sous problèmes qui font par leurs complexité une branche à part a traité, d'un côté les agents doivent être dotés d'un mécanisme d'exploration (marche aléatoire, suivi de chemin ...), de mécanismes leurs permettant la localisation de la base et la détection de ressources, des mécanismes de coordination et de coopération pour augmenter la performance du groupe.

Notre objectif est de construire des systèmes composés d'un ensemble d'agents assurant un contrôle décentralisé pouvant améliorer le système à un tout cohérent. Plusieurs démarches sont envisageables pour la construction de tels systèmes ; Ceux qui ont envahi les recherches dans cette décennie sont les démarches qui s'inspirent des systèmes collectifs observés dans la nature comme métaphore pour proposer de nouveaux mécanismes ; ces méthodes sont aussi intéressante, mais nécessite un travail important de la part du concepteur. On cherche donc à concevoir un SMA réactif coopératif, caractérisé par un contrôle décentralisé, où chaque agent n'a qu'une vision partielle de l'environnement dans lequel il est plongé. Les actions qu'il peut émettre ne peuvent modifier l'environnement que localement ; et chaque agent décide localement des actions à entreprendre, en suivant ses perceptions reçues par ses capteurs ; mais tous les agents contribuent à l'évolution du système.

Dans ce mémoire on va se concentré sur un algorithme qui a été proposé dans le cadre de la résolution de ce problème, c'est l'algorithme de MRSAM (Sarid, 2011), certains amélioration ont été attribué à ce dernier algorithme pour assurer plus de recrutement d'agents dans des chemins colorés, dont on appelé la nouvelle version d'algorithme : (SMRSA) Spiral Multi Robot Search Area , un ensemble énorme de simulation a été fait et les résultats montre que nos améliorations donnent des meilleurs résultats par rapport à celui de MRSAM en ce qui concerne le temps de résolution. Enfin plusieurs problèmes ont été aussi rencontrés et qui font l'objet de nos perspectives.

Plan de mémoire :

Ce manuscrit se décompose en quatre chapitres :

Le premier chapitre : constitue un état de l'art, qui traite le concept agent et tout ce qui tourne autour, et les systèmes multi agents et les concepts fondamentaux qui ont découle.

Le deuxième chapitre : couverture et exploration des environnements inconnus, ce chapitre se divise en deux parties, la première partie traite quelques algorithmes de couverture et ses principes de recherche d'une cible dans un environnement inconnu. et la deuxième partie traite le problème d'exploration.

Le troisième chapitre : conception et réalisation la conception proposé qui se résume à un schéma de comportement d'agent coopératif et réactif ainsi que l'algorithme utilisé pour résoudre le problème de fourragement ainsi que les résultats expérimentaux sont discutés dans ce chapitre, la suite des simulations réalisé montre que nous amélioration de l'algorithme MRSAM (Sarid, 2011) qui se résume par notre algorithme (SMRSA), donnent des bonnes résultats par rapport à l'algorithme original. Un ensemble des simulations a été aussi fait pour tester quelle sont les paramètres (nombre des régions, nombre d'agent...) qui peuvent affecter le temps de résolution et la robustesse de notre algorithme aux bruit.

Enfin nous Terminerons le mémoire avec une conclusion générale et quelle que perceptives.



chapitre 1

Etat de l'art

Introduction

Depuis quelques années, les systèmes multi-agents ont pris une place de plus en plus importante en informatique, que ce soit dans le domaine de l'intelligence artificielle, dans ceux des systèmes distribués, de la robotique, ou même dans la "vie artificielle". Les recherches dans le domaine des systèmes multi-agents poursuivent deux objectifs majeurs: le premier concerne l'analyse théorique et expérimentale des mécanismes d'auto-organisation qui ont lieu lorsque plusieurs entités autonomes interagissent; le second s'intéresse à la réalisation d'artefacts distribués capables d'accomplir des tâches complexes par coopération et interaction.

Le problème de recherche d'une cible dont sa position est inconnue à l'avance est très important dans plusieurs applications du monde réel et des recherches académiques. Dans ce problème, la couverture de région est une tâche correspondante parce que la recherche jusqu'à cette cible, va engendrer une couverture d'une certaine région avant d'arriver à cette cible, c'est pour cela le problème est considéré comme un problème de couverture partielle d'une région.

Ce chapitre est divisé en deux parties d'états de l'art. La première partie concerne tout ce qui est en relation avec les notions théoriques des SMA, à savoir l'IAD, les agents, l'interaction, coordination, coopération et simulation. La deuxième partie, se concentre plutôt sur le problème de couverture où on cite les travaux réalisés dans la littérature et se divise en deux : couverture mono-robot et couverture multi-robots.

I. Systèmes multi-agents: définition et principes

Les systèmes multi-agents font partie d'Intelligence Artificielle Distribuée, qui, à la différence d'IA classique, s'intéresse aux comportements intelligents, résultant de l'activité coopérative de plusieurs agents. *Le système multi-agents est considéré comme un système distribué composé d'un certain nombre d'entités autonomes (les agents), qui travaillent selon des modes complexes d'interaction, pour réaliser leurs propres buts et par-là même atteindre l'objectif global désiré. Les agents peuvent interagir en communiquant directement entre eux ou par l'intermédiaire d'un autre agent ou en agissant sur leur environnement.* [Gouasmi ,05]

I.1. l'intelligence artificielle distribuée:

L'intelligence artificielle est reconnue comme étant une discipline informatique qui a pour objectif de modéliser ou de simuler des comportements humains dits intelligents tels que la perception, la prise de décision, l'apprentissage, etc. mais l'IA a vite rencontré un certain nombre de difficultés, dues pour la plupart à la nécessité d'intégrer, au sein d'une même base de connaissances, l'expertise, les compétences et les connaissances d'individus différents qui, dans la réalité, communiquent et collaborent à la réalisation d'un but commun.

À la différence de l'Intelligence Artificielle classique qui modélise le comportement intelligent d'un seul agent, l'intelligence artificielle distribuée s'intéresse à des comportements intelligents qui sont le produit de l'activité coopérative de plusieurs agents.

Le passage du comportement individuel aux comportements collectifs est considéré non seulement comme une extension mais aussi comme un enrichissement de l'IA, d'où émergent de nouvelles propriétés et de nouveaux comportements. [Labidi et al, 93]

I.2. Agents : concepts fondamentaux

Le terme « agent » n'est pas aisée du fait du caractère interdisciplinaire des agents qui, d'un côté, sont sujets aux effets des différentes directions de recherches scientifiques et, de l'autre, reflètent les besoins d'applications pratiques. Ainsi, il est difficile de faire accepter par tous une définition précise de cette notion. Car le terme agent étant un terme actuellement porteur, il est utilisé de manière assez vague.

I.2.1. Définition

Définition 1 : Agent [Ferber, 95]

Un agent est une entité :

- qui est capable de percevoir (de manière limitée) son environnement,
- qui ne dispose que d'une représentation partielle de son environnement,
- qui est capable d'agir dans son environnement,
- qui peut communiquer directement avec d'autres agents,
- qui possède des ressources propres,
- qui possède des compétences et peut offrir des services,
- qui est mue par un ensemble de tendances (sous forme d'objectifs individuels),

- dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception et de ses représentations.

A titre d'exemple, on peut dire que l'homme et la fourmi sont des agents physiques. En ce qui concerne l'aspect logiciel, une définition synthétique peut être donnée

Définition 2 : [Wooldridge, 97]

"Un agent est une entité qui exhibe les caractéristiques suivantes : autonomie, réactivité, proactive flexible et qui possède une certaine habilité sociale." [Gouasmi, 05]

Ou :

- **Autonome :** Son comportement est fonction de ses perceptions qui agisse sur son état, et de sa représentation de l'environnement dans lequel il évolue.
- **Réactivité :** un agent réactif maintient un lien constant avec son environnement et répond aux changements qui y surviennent.
- **Proactif:** Il peut prendre des initiatives afin de satisfaire ses buts. Pour se faire, il n'est pas soumis à l'invocation d'une autre entité pour agir mais peut agir sur sa propre initiative.
- **Flexible:** Il adapte son comportement à sa perception de son environnement et peut participer à des organisations afin de mieux satisfaire son but
- **Social:** Il a la capacité d'interagir pour atteindre ses buts ou pour aider d'autres agents dans leurs activités.

I.2.2. Classification d'agent :

A partir de ces définitions, différents modèles d'agents peuvent être envisagés suivant deux axes : la vision fonctionnelle et la vision décisionnelle :

- La vision fonctionnelle s'appuie sur le degré d'intégration des facettes autonomie, environnement, interaction et organisation dans l'agent. Un agent est autonome, interagissant ou social.

- La vision décisionnelle s'appuie sur le degré de couplage à l'environnement. Un agent est réactif, hybride ou cognitif.

Ces deux visions sont orthogonales et, par conséquent, complémentaires et, les propriétés et caractéristiques des agents sont fortement liées à la représentation de l'environnement. Ainsi, plus un agent possède une représentation élaborée de son environnement, plus il lui est possible d'avoir une activité cognitive [Champion ,03]

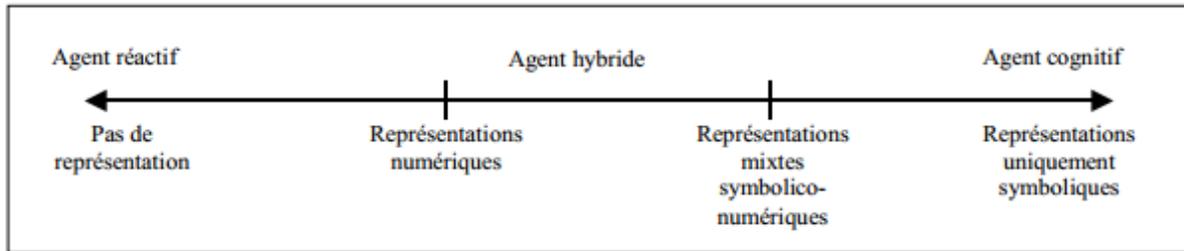


Figure 1.1 : Axe pratique d'évaluation de la capacité d'un agent à accomplir individuellement des tâches complexes et à planifier ses actions [Champion ,03]

- **Agent cognitif :**

Les agents cognitif disposent d'une base de connaissances comprenant les diverses informations liées a leurs domaines d'expertise et a la gestion des interactions avec les autres agents et leur environnement. Les agents sont généralement intentionnels c'est-à-dire qu'ils possèdent des buts et des plans explicites leur permettant d'accomplir leurs buts. [Courdier, 07]

- **Agent Réactif :**

Les agents réactifs sont souvent qualifiés de ne pas être " intelligents " par eux-mêmes. Ils sont des composantes très simples qui perçoivent l'environnement et sont capable d'agir sur celui-ci. Ils n'ont pas une représentation symbolique de l'environnement ou des connaissances et ils ne possèdent pas de croyances, pas de mécanismes d'envoi de messages. Leurs capacités répondent uniquement au mode stimulus/action qui peut être considéré comme une forme de communication. Un SMA constitué d'agents réactifs possède généralement un grand nombre d'agents et présente un comportement global intelligent. [Haddad et Hamidi, 09]

- **Agent hybride :**

Les agents hybrides sont conçus pour combiner des capacités réactives à des capacités cognitives, ce qui leur permet d'adapter leur comportement en temps réel à l'évolution de l'environnement.

Un agent hybride consiste en la combinaison de plusieurs caractéristiques au sein d'un même agent ; ces caractéristiques concernent la collaboration, l'autonomie et la capacité à apprendre ..., etc. De façon habituelle, un agent n'est pas complètement de type réactif ni de type délibératif. C'est plutôt une combinaison de ces deux approches. Dans certaines circonstances, un agent doit agir très rapidement alors que dans d'autres, il aura la possibilité de prendre plus de temps pour avoir un processus délibératif de meilleure qualité. Raison pour laquelle, ils sont dits hybrides [Haddad et Hamidi, 09].

Les architectures hybrides entre les deux approches, verticales et horizontales, il existe tout un éventail d'architectures hybrides qui fait intervenir un modèle modulaire avec une approche centralisée ou décentralisée de la gestion des composants d'un agent. L'agent est alors conçu comme une association de modules plus ou moins spécialisés qui fonctionnent en parallèle et qui interagissent. Les interactions entre ces modules permettent d'obtenir un comportement global cohérent. Cette approche permet de combiner des composantes réactives avec des composantes cognitives [Haddad et Hamidi, 09].

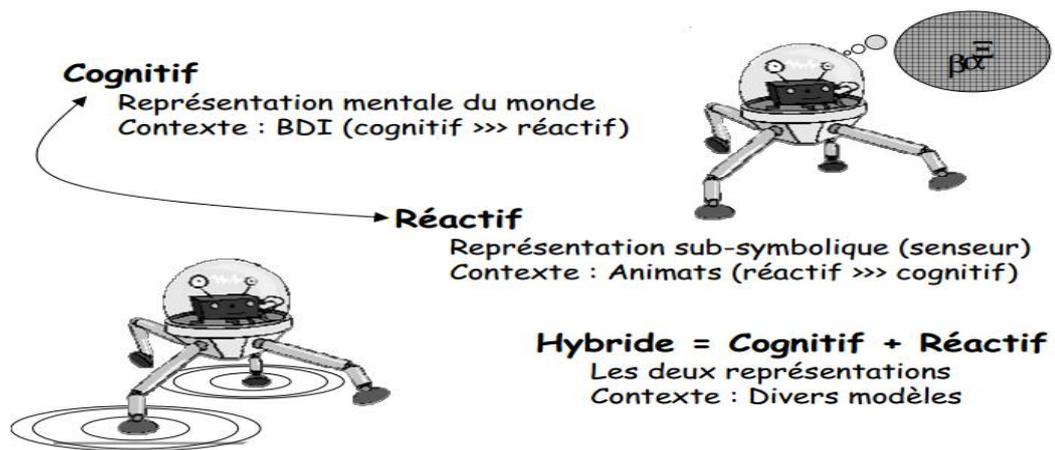


Figure 1.2 : Classification d'agents [Courdier ,07]

I.2.3. Architecture d'agent

Un agent se caractérise essentiellement par la façon dont il est conçu et par ses actions, en d'autres termes par son architecture et son comportement. L'architecture correspond à un point de vue de concepteur, qui peut se résumer ainsi : comment assembler les différentes parties d'un agent de manière qu'il accomplisse les actions que l'on attend de lui ? L'architecture d'un agent caractérise ainsi sa structure interne, c'est-à-dire le principe d'organisation qui sous-tend l'agencement de ses différents composants. L'architecture d'un agent est donc une organisation interne modulaire manipulant des flux de données (diffusion, traduction) et gérée par une structure de contrôle (par inhibition, hiérarchique...). [Champion, 03]

- **Architecture BDI pour un agent cognitif :**

Une architecture BDI est conçue en partant du modèle "**Croyance-Désir-Intention**", en anglais "Belief-Desire-Intention", de la rationalité d'un agent intelligent. [Adina, 02]

Le B = Belief = Croiance : Les croyances d'un agent sont les informations que l'agent possède sur l'environnement et sur d'autres agents qui existent dans le même environnement.

Le D = Desire = Désir : Les désirs d'un agent représentent les états de l'environnement, et parfois de lui-même, que l'agent aimerait voir réalisés.

Le I = Intention = Intention : Les intentions d'un agent sont les désirs que l'agent a décidé d'accomplir ou les actions qu'il a décidé de faire pour accomplir ses désirs.

La figure suivante présente les composantes principales d'une architecture BDI.

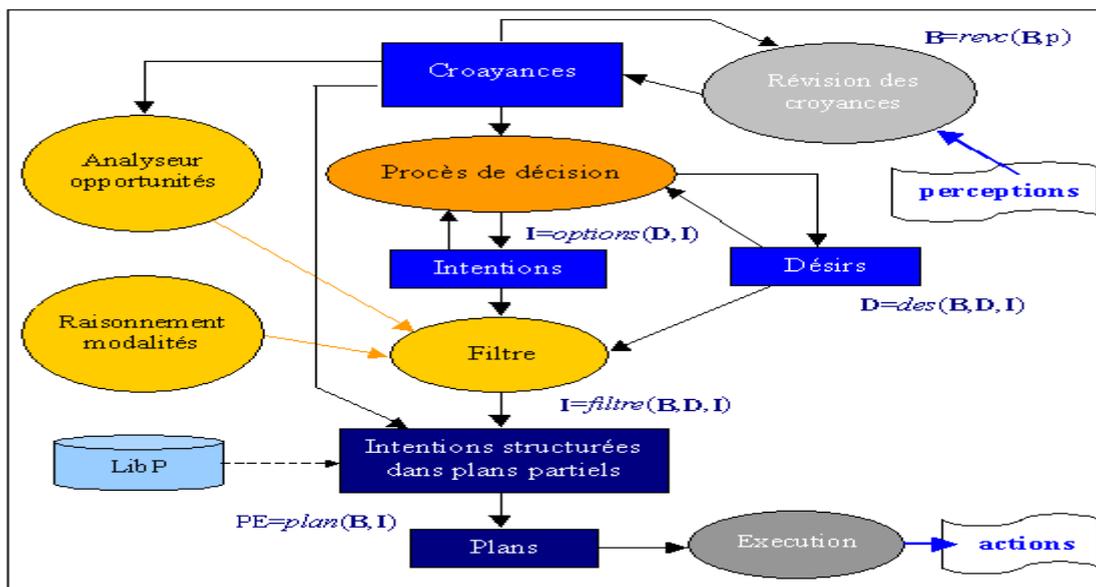


Figure 1.3: Architecture BDI. [Adina, 02]

- **Architecture de subsomption pour les agents réactifs [Brooks 1990]:**

Les architectures verticales sont des architectures adaptées aux agents réactifs. Elles caractérisent des agents modélisés en termes de fonctions de transition entre les données perçues et les actions exécutées. Un principe de décomposition verticale revient à associer à chaque action une fonction simple, elle offre l'avantage d'une conception simple, car ne faisant intervenir que trois composantes : perception-sélection-action [Haddad et Hamidi, 09].

Les architectures réactives représentent le fonctionnement de l'agent au moyen de composantes avec une structure de contrôle simple, et sans représentation évoluée des

connaissances de l'agent. L'intelligence de l'agent est vue comme étant le résultat des interactions entre ces composantes et l'environnement. Cela veut dire qu'une telle architecture peut résoudre des problèmes complexes, qui normalement demandent un comportement intelligent, sans traiter l'intelligence du point de vue classique de l'intelligence artificielle.

[Adina, 02]

L'architecture subsumption [Brooks, 91] est l'une des architectures verticales les plus connues et utilisées dans le domaine de la robotique. Cette architecture intercale entre la perception et d'action couches de comportement de complexité croissante : sur la couche basse, les comportements de base tels que le mouvement aléatoire ; sur les couches les plus hautes ceux devant permettre, théoriquement, de satisfaire des comportements spécifiques. Les couches fonctionnent en parallèle et accèdent à un sous ensemble des informations captées. L'accès au système d'action est contrôlé via une hiérarchie de contrôle : les comportements présents sur une couche peuvent supplanter les comportements présents sur les couches inférieures en substituant leurs propres entrées aux entrées des couches inférieures en inhibant tout sortie de ces mêmes couches pour des périodes de temps préprogrammées. À part ces signaux descendants de suppression et d'inhibition, aucun autre type de communication n'a lieu entre les couches [Briot et Demazeau, 01].

L'architecture réactive la plus connue et la plus influente est celle proposée par Rodney Brooks ; elle s'appelle **architecture de subsumption**, en anglais "subsumption architecture".

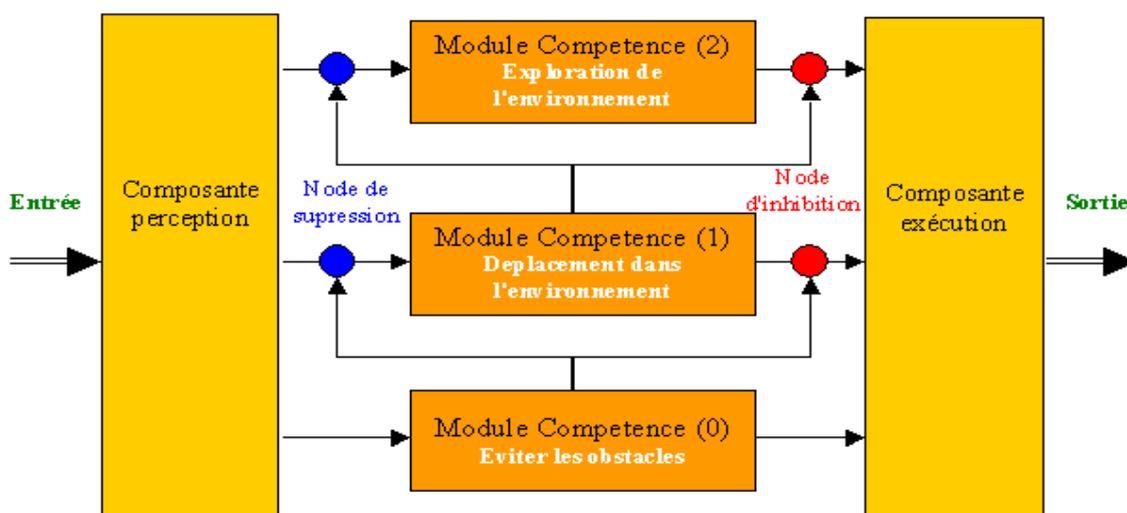


Figure 1.4 : Architecture subsumption. [Adina, 02]

Un système avec un seul agent est de l'intelligence artificielle classique ; un système avec les agents multiples est une société artificielle. Donc, les thèmes principaux et les fondations de l'intelligence artificielle distribuée sont l'organisation, la coordination et la coopération [Sahki, 08]

I.3. Principes des SMA

Un système multi-agents comporte plusieurs agents qui interagissent entre eux dans un environnement commun. Certains des agents du SMA sont des agents logiciels, certains agents peuvent être des personnes, ou même des machines mécaniques.

Il est composé des éléments suivant :

- Un environnement **E** : c'est-à-dire un espace disposant généralement d'une métrique.
- Un ensemble d'objets **O** : Ces objets sont situés, c'est-à-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans **E**. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
- Un ensemble **A** d'agents : qui sont des objets particuliers (**A** est contenu dans **O**), lesquels représentent les entités actives du système.
- Un ensemble **R** de relations : qui unissent des objets (et donc des agents) entre eux.
- Un ensemble d'opérateurs **Op** : permettant aux agents de **A** de percevoir, produire, consommer, transformer et manipuler des objets de **O**.
- Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on pourrait appeler « les lois de l'univers ». [Courdier, 07]

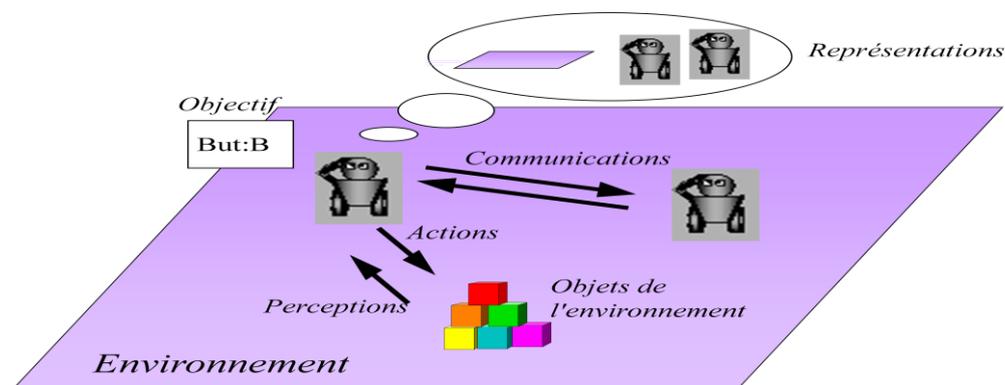


Figure 1.5 : Les composantes principales d'un SMA

Nous présentons dans cette section les composantes d'un SMA pour la création d'un collectif d'agents: l'interaction, l'organisation, la communication, et la coopération

I.4. Interactions dans les SMA

La notion d'interaction constitue l'essence d'un système multi-agents puisque c'est grâce à elle que les agents vont pouvoir produire des comportements collectifs complexes et dépendants les uns des autres. En effet, la fonction interactionnelle d'un agent porte sur l'ensemble des mécanismes lui permettant de faire le lien avec ce qui l'entoure (son environnement ainsi que l'ensemble des autres agents). L'interaction peut être vue comme une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques [Thomas 05]. Elle représente de plus un élément nécessaire à la constitution des organisations.

« On appelle situation d'interaction un ensemble de comportements résultant du regroupement d'agents qui doivent agir pour satisfaire leurs objectifs en tenant compte des contraintes provenant des ressources plus ou moins limitées dont ils disposent et de leurs compétences individuelles » [Ferber 95].

En effet, un des aspects importants de la dynamique d'un SMA est la nature des interactions entre ses entités, puisqu'elles constituent un support effectif de la coopération [Panait et al. 05]. Les agents interagissent en vue de coopérer et de coordonner leurs actions afin d'atteindre des buts locaux (individuels) ou globaux (collectifs). L'interaction est réalisée à travers un langage compréhensible et commun à tous les agents et peut être :

- sélective sur un nombre restreint d'agents (par exemple en fonction des accointances c'est-à-dire les agents qu'il connaît) ;
- ou étendue à l'ensemble des agents c'est-à-dire par diffusion.

De plus, les interactions permettent dans les systèmes multi-agents la combinaison des fonctionnalités des agents pour faire émerger le comportement global du système.

[Sabouret, 09]

D'autre part, une interaction du point de vue d'un agent peut être dans à l'une des trois catégories suivantes selon le gain perçue :

- incidence nulle ;
- incidence positive : l'agent perçoit l'interaction comme une aide ;
- incidence négative : l'agent perçoit l'interaction comme une gêne.

Les interactions entre agents peuvent variées selon les situations dont se trouvent ces agents : coexistence, compétition ou coopération.

S'ils ne font que coexistence, alors chaque agent ne considère les autres agents que comme des composantes de l'environnement. Il n'y a aucune communication directe entre les agents.

S'ils sont en compétition, alors le but de chaque agent est de maximiser sa propre satisfaction. La compétition entre agents peut avoir plusieurs sources : Les buts des agents peuvent être incompatibles ou les ressources peuvent être insuffisantes.

S'ils sont en coopération, alors le but des agents n'est pas seulement de maximiser leurs propres satisfactions mais aussi de contribuer à la réussite de groupe. Les agents travaillent ensemble à la résolution d'un problème commun.

La coopération est une caractéristique très importante des SMA. [Sahki, 08] En effet, une résolution distribuée d'un problème ou le partage des tâches et de résultats est les résultats d'interactions coopératives entre les différents agents qui disposent d'un ensemble des caractéristiques qui leur rendent coopératifs.

I.5. Organisations dans les SMA

Lorsqu'on parle d'organisations on suppose l'existence d'un ensemble d'entités en relation formant une unité. Dans un système multi-agents, l'organisation permet aux agents de savoir quels sont leurs partenaires et quels rôles ils jouent de façon à répondre à un objectif donné. C'est un arrangement des agents et de leurs comportements conditionné par les contraintes imposées par l'environnement [Ferber et al, 2009], [Deloach, 2009]. C'est une structure du système [Jiquan et al, 2008]. Une organisation est un ensemble d'individus regroupés au sein d'une structure régulée, ayant un système de communication pour faciliter la circulation de l'information, dont le but est de répondre à des besoins et d'atteindre des objectifs déterminés [Moujahed, 07].

Dans un système multi-agents, l'organisation est le facteur structurant, elle permet aux agents de savoir quels sont leurs partenaires et quels rôles ils jouent de façon à répondre à un objectif donné. Ferber [Ferber , 95] insiste sur la dualité entre l'aspect statique et l'aspect dynamique de l'organisation en précisant que « l'organisation est à la fois le processus d'élaboration d'une structure et le résultat même de ce processus ».

Précisent que l'organisation est le résultat d'un comportement social des agents et elle ne peut être créée explicitement par le concepteur.

Les communications dans les SMA comme chez les humains sont à la base des interactions et d'organisation.

I.6. Communication dans les systèmes multi-agents :

Sans communication, l'agent n'est qu'un individu isolé. C'est parce que les agents communiquent qu'ils peuvent coopérer, coordonner leurs actions et réaliser des tâches en commun.

I.6.1. Définition par Russel [Russel, 06] comme :

« La communication est l'échange intentionnel d'informations occasionné par la production et la perception de signes issus d'un système partagé de signes conventionnels. »

Les communications dans les SMA peuvent être classées en deux classes distinctes, à savoir communications directes et indirectes.

- **Communications indirectes (ou implicites) :** Ce sont des mécanismes fondés sur l'usage de signaux qui prennent leurs sources en biologie et en éthologie. En effet, les agents dans de tels systèmes peuvent laisser des traces (signaux) de leur présence ou de leur action qui sont perçues et interprétées par d'autres agents (tels que les phéromones chez les fourmis). Dans d'autres systèmes, les agents dans l'environnement émettent des signaux ou des champs de potentiels guidant les autres agents dans un but de compétition ou de coopération (tels que les agents robots qui peuvent émettre des signaux pour attirer ou repousser d'autres agents de leur environnement).
- **Communications directes (ou explicites) :** Ce modèle de communication entre agents repose sur la communication par message. L'échange direct est réalisé volontairement en direction d'un individu ou d'un groupe d'individus. Dans ce type de communication les agents sont en liaison directe, les messages sont envoyés directement et explicitement de l'émetteur au destinataire et ce via un canal de communication.

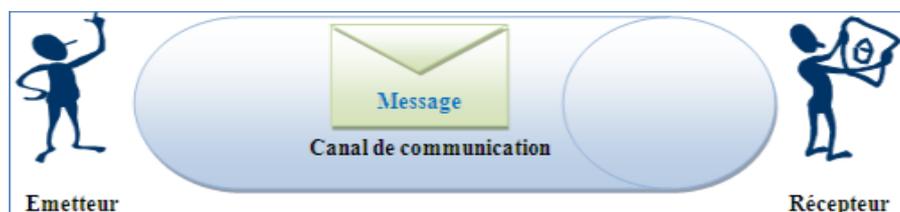


Figure 1.6 : Communication directe [Boussebough, 11].

I.7 Coopération

Les protocoles de coopération à décomposer un problème en tâches puis à les distribuer. Le système doit choisir parmi les décompositions possibles et doit considérer les ressources et les possibilités de chacun des agents. [Durfee & Montgomery, 90]

La décomposition peut être effectuée par le concepteur, par les agents en utilisant une planification hiérarchique ou être directement liée à la nature de problème. Une fois la décomposition réalisée, il faut distribuer les tâches selon les critères suivants :

- Éviter le sur-chargement des ressources critiques.
- Assigner les tâches aux agents ayant des capacités correspondantes.
- Laisser un agent avec grand champ d'observation assigner les tâches des autres agents.
- Donner des responsabilités qui se chevauchent aux agents dans un souci de cohérence.
- Assigner des tâches interdépendantes à des agents proches spatialement ou sémantiquement pour limiter les coûts de communication et de synchronisation.
- Réassigner les tâches pour accomplir les plus urgentes.

On utilise les systèmes multi-agents pour simuler des interactions existant entre agents autonomes. On cherche à déterminer l'évolution de ce système afin de prévoir l'organisation qui en résulte. Par exemple, en [sociologie](#), on peut paramétrer les différents agents composant une communauté.

I.8. Les mécanismes de coordination multi-agent

Le mécanisme spécifique de coordination qui est actif dans un SMA détermine sa dynamique interne, i.e. les interactions entre les agents ainsi que les propriétés externes de la société d'agents. Les mécanismes de coordination varient avec les domaines dans lesquels ils sont utilisés et en fonction de la propension des agents à coopérer, de l'environnement et de son degré de stabilité et de prédictibilité, ainsi que des tâches à réaliser.

Pourquoi coordonner ?

Appliquer la définition de la coordination au cadre des SMA amène à dire que la coordination est la gestion des dépendances qui existent entre les activités des agents. Or, dans un SMA, les activités des agents sont souvent incomplètement spécifiées car il est impossible d'avoir des agents omniscients disposant d'informations complètes et à jour sur l'état complet du système dont ils font partie. Dans ce contexte, la coordination correspond avant tout aux différents processus ou actions que les agents doivent engager afin de réaliser leurs actes sociaux de manière cohérente : l'unité globale du système issue des comportements locaux des agents doit être préservée. Sans coordination, un système est voué au désordre.

Comment coordonner ?

Au niveau d'un SMA, la coordination nécessite de déterminer l'existence de situations d'interaction, d'identifier ces situations d'interactions et de les gérer.

La question est de savoir comment mettre en oeuvre une coordination au niveau du système.

De nombreuses questions de recherches se posent alors :

- Les mécanismes de coordination doivent-ils être implémentés au sein des agents, entre les agents, au travers de l'environnement, au travers de l'organisation, au travers des échanges ?
- Si l'on décide d'associer un mécanisme au niveau des agents, doit-on diminuer les interactions entre les agents en les spécialisant, en réduisant leurs interdépendances ou augmenter les capacités locales afin de prendre des décisions localement plus adaptées ?
- Si l'on décide d'associer un mécanisme au niveau des organisations : doit-on spécialiser les agents et former des structures organisationnelles centralisées, hiérarchiques, de marchés ou de communautés ?
- Si l'on décide d'associer un mécanisme au niveau des interactions : doit-on faire échanger des informations afin d'augmenter les connaissances sur le système (représentation des autres, méta-information, abstractions) et gérer les communications (mesure d'intérêts, de temps de réponse...) ? [Champion ,03]

Toutes ces possibilités de mise en oeuvre de mécanismes de coordination sont envisageables et peuvent être combinées pour donner naissance à d'autres mécanismes [Bond et al, 88]. Les mécanismes multi-agents les plus courants et occupant une place privilégiée dans la littérature sont présentés par la Figure 1.7. Ces mécanismes multi-agents sont issus des réflexions menées sur la classification (selon le type des dépendances à gérer) des mécanismes théoriques de coordination et appliquées au contexte des systèmes multi-agents.

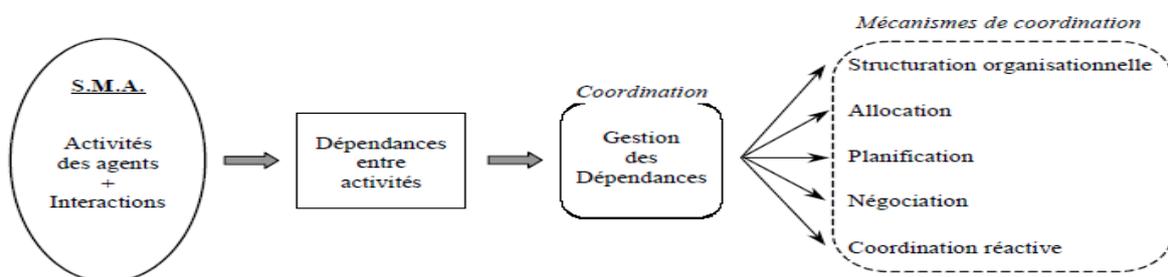


Figure 1.7 : Principaux mécanismes de coordination multi-agent.

Ces mécanismes sont maintenant exposés un à un dans les parties suivantes. Il s'agit de la structuration organisationnelle, de l'allocation (de ressources ou de tâches), de la planification, de la négociation et de la coordination réactive.

I.8.1. Structuration organisationnelle

L'organisation définit implicitement les responsabilités, les capacités et le rapport à l'environnement des agents. La structuration organisationnelle exploite cette structure organisationnelle prédéfinie du SMA qui fournit un cadre pour les activités et les interactions à travers la définition des rôles, des moyens de communication et des relations d'autorité. Dans un SMA, l'organisation est habituellement vue comme l'ensemble des relations à long terme entre agents. Quand les agents jouent certains rôles dans une organisation, ils agissent suivant le comportement que ces rôles et leurs relations impliquent. [Champion ,03]

Il existe deux configurations extrêmes de ce type d'organisation. Dans une organisation *hiérarchique*, chaque tâche est contrôlée par un agent gestionnaire qui coordonne son processus de réalisation [Durfee et al , 89]. A l'opposé, dans une organisation *latérale*, il n'existe pas de gestionnaire unique et chaque tâche est réalisée de manière coopérative. La configuration intermédiaire et la plus courante d'organisation est la structure informelle dans laquelle cohabitent une certaine hiérarchie induite par des relations d'autorité, des contacts en diagonale et des contacts horizontaux directs entre les agents (Figure 1.8).

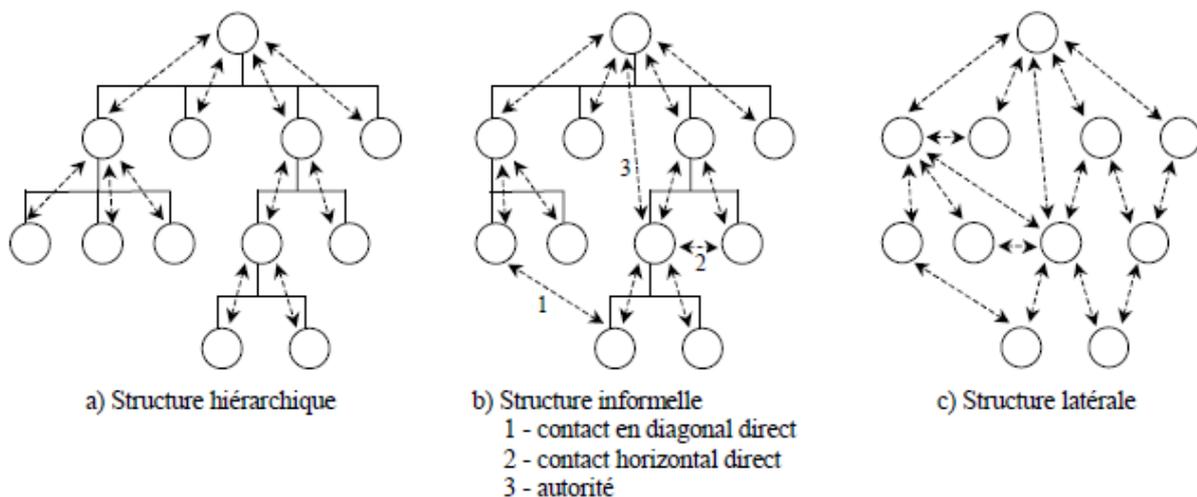


Figure 1.8 : Exemples de structures organisationnelles (d'après [Ossowski 98])

I.8.2. Allocation

La résolution de problèmes s'effectue généralement en quatre phases : la décomposition des tâches (et/ou des ressources), l'allocation des tâches (et/ou des ressources), l'exécution des tâches (et/ou des ressources) et la synthèse des résultats de l'exécution. Dans ce contexte, l'allocation essaie de répondre à la question : qui doit faire quoi et comment, en fonction des

but, des compétences des agents et des contraintes contextuelles, sachant que les agents impliqués pouvant être différents les uns des autres

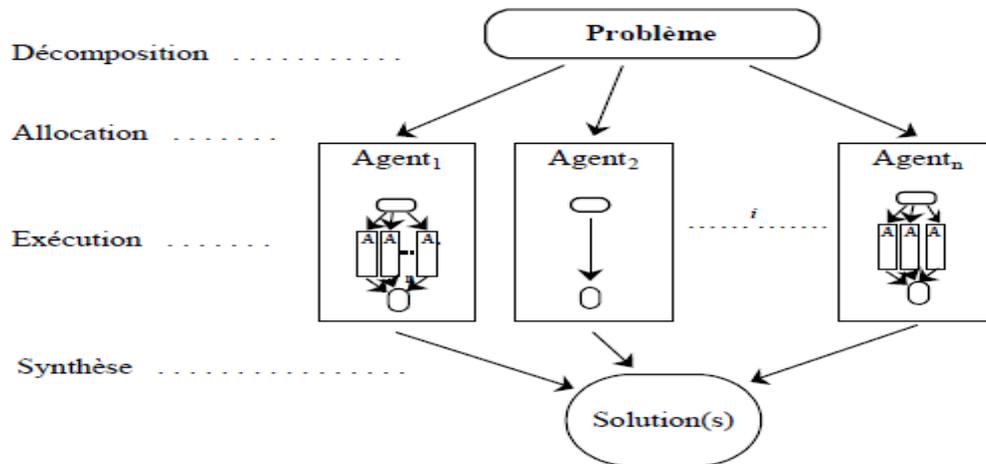


Figure 1.9: Résolution d'un problème par un SMA.

Concernant l'allocation de tâches, celle-ci peut s'effectuer de deux manières selon qu'elle dépende d'un ou de plusieurs agents. D'une part, si un seul agent décompose un problème en sous-problèmes et répartit les tâches entre les autres agents, l'allocation est centralisée. D'autre part, si chaque agent est capable de décomposer son problème en sous problèmes et de répartir les tâches associées, l'allocation est distribuée. La méthode d'allocation de tâches la plus étudiée est probablement le réseau contractuel Il s'agit d'une allocation distribuée et L'allocation de ressources répond à une problématique similaire à celle de l'allocation de tâches et celles-ci sont souvent liées. Les mécanismes employés pour l'allocation de ressources sont souvent fondés sur des systèmes issus de l'économie tels que le contrôle de marché ou les enchères.

I.8.3. Planification

La planification multi-agent tente de coordonner les actions de *plusieurs agents* de manière à ce qu'un but commun soit atteint (exemples classiques : la construction d'une maison, l'assemblage de composants...) [Hendler, 90]. Du fait que les agents établissent des plans qui spécifient leurs actions et interactions futures en fonction d'un certain objectif, tous les agents impliqués dans la planification s'entendent pour se comporter suivant ces plans.

La planification dans un SMA peut se décomposer en trois étapes : la construction de plans, la synchronisation de plans (coordination) et l'exécution de plans. Comme il est possible d'utiliser un ou plusieurs agents à chacune de ces étapes, il est possible d'obtenir un grand nombre d'organisations différentes. Un système de planification est alors composé d'un

ensemble d'agents pouvant planifier, synchroniser ou exécuter des plans, un même agent pouvant accomplir une seule ou plusieurs de ces tâches

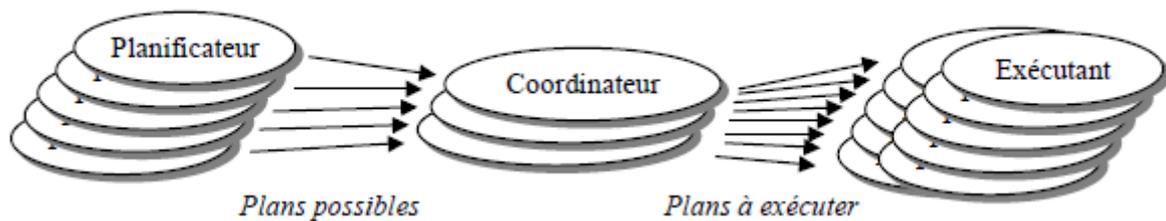


Figure 1.10 : Système de planification multi-agent (exemple).

I.8.4. Négociation

Une part importante du travail effectué sur la coordination multi-agent concerne la négociation car la majorité des mécanismes de coordination peut entraîner un processus de négociation. La négociation peut sembler être la solution ultime pour la résolution de conflit et la littérature est prolixe concernant tout ce qui s'y rapporte.

Pour définir la négociation, il est possible de dire qu'il s'agit d'un processus par lequel une décision conjointe est prise par deux ou plusieurs parties.

La négociation comporte quatre aspects : la décision, le langage, l'objet de la négociation et le processus [Müller, 96] :

- La décision : Il s'agit de la dimension intra-agent, elle est relative à la modélisation du raisonnement du négociateur. Pour prendre une décision quant au déroulement d'une négociation, l'agent peut avoir recours à des *fonctions d'utilité* (utilisation de la théorie des jeux, par exemple), à des fonctions de comparaison et de corrélation, à des préférences et à des *stratégies de négociation*. La stratégie de négociation occupe une place prépondérante dans le développement d'un processus de négociation. Une stratégie détermine quelle option, parmi celles permises, un agent décide de choisir. Une stratégie peut relever d'un comportement purement égoïste ou, plus rarement, altruiste de la part du négociateur. Elle est, dans tous les cas, basée sur des critères locaux, non contraints par des conventions externes.

- L'objet de la négociation : Le point central d'un modèle de négociation est de représenter le sujet pour lequel il existe un différent parmi les agents. L'objet de la négociation est fortement dépendant du contexte et peut être de nature diverse (tâches, intervalles de temps, ressources, plans, etc.).

- Le langage : Il s'agit de la dimension inter-agent, elle est relative à la communication

(explicite ou implicite) entre les agents. Comme il n'est pas toujours suffisant de communiquer l'*objet* de la négociation, un agent a besoin d'exprimer comment l'autre partie devrait interpréter un message. Pour répondre à cette problématique, des travaux ont été réalisés à partir de la théorie des actes de langage et ont mené à l'élaboration de *primitives* de négociation. Enfin, un *protocole* de négociation doit être défini afin de déterminer la séquence des messages. Le protocole est généralement une convention acceptée et utilisée par toutes les parties impliquées dans le processus, il s'agit d'une connaissance commune. Le protocole fait référence aux primitives et contraint l'objet de la négociation.

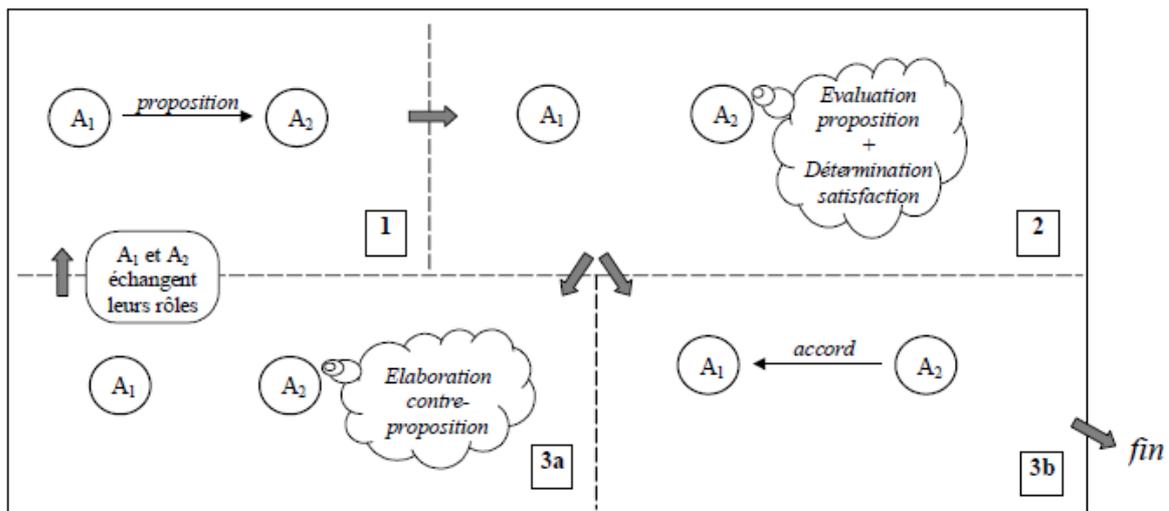


Figure 1.11 : Exemple de protocole de négociation entre deux agents.

- Le processus : Il s'agit de la dimension globale, elle est relative au comportement du système. Le processus est fonction des modèles de négociation mis en œuvre et des comportements des participants. Il peut prendre plusieurs formes. Lorsqu'il est question d'une *négociation par compromis*, chacune des parties relâche les contraintes les moins importantes et l'accord est établi lorsque toutes les contraintes sont satisfaites. Lorsqu'il est question d'une *négociation intégrante*, les agents cherchent à identifier les buts profonds et à trouver une solution qui satisfait complètement ces buts. [Champion, 03]

I.8.5. Coordination réactive

La coordination réactive s'intéresse essentiellement, comme son nom l'indique, à la coordination des agents réactifs. Ceux-ci étant très simples de par leur structure interne et possédant une représentation très limitée de leur environnement, des mécanismes de coordination dédiés ont du être mis en œuvre. Plusieurs cas sont possibles.

Si les agents ont des buts indépendants, les seuls problèmes qu'ils peuvent rencontrer viennent des conflits éventuels d'accès à des ressources communes. Il est alors possible de distinguer

deux grandes catégories de situations. Les premières ont trait à l'accès d'une même portion d'espace par un ensemble d'agents. Il s'agit alors pour eux d'accomplir leurs buts tout en s'évitant. Les techniques utilisées sont alors des techniques définissant des comportements répulsifs et attractifs tels que le comportement de meute [**Reynolds, 87**] (pour constituer des meutes de robots [**Mataric, 92**]) ou le comportement de bancs de poissons. Une des techniques les plus abouties est celle dite des champs de vecteurs, utilisée pour la résolution de conflits relatifs aux trajectoires de vol des avions [**Zeghal, 93**]. Les secondes catégories de situations portent sur l'utilisation d'une ressource ponctuelle, telle un outil commun. Il est alors possible d'utiliser des mécanismes de synchronisation.

Si les agents ont des buts dépendants et que les actions des uns peuvent améliorer celles des autres et donc augmenter les performances du système dans son entier, le principe général consiste à utiliser les capacités des agents réactifs à réagir aux modifications de l'environnement et souvent à marquer cet environnement pour coordonner les actions des agents entre eux. Les exemples classiques sont la récupération de minerais par des robots explorateurs et la résolution de problèmes par création de « chemins d'odeur », métaphore faisant référence aux fourmis et aux phéromones qu'elles laissent sur leur passage [**Drogoul, 93**].

I.9. Simulation multi agent

La simulation multi-agent permet une validation d'un modèle. La structure, la dynamique et l'évolution d'un système sont liées. Lorsqu'il s'agit d'imiter une réalité à l'aide de systèmes multi-agents, modèle et simulation sont liés.

En simulation multi-agent, la vision d'un phénomène est, si l'on considère que celui-ci est modélisé suivant les principes des systèmes multi-agents, distribué. Ainsi, le phénomène réel à simuler est décomposé (naturellement ou volontairement) en un ensemble d'éléments discrets autonomes dont les interactions reproduisent le phénomène. Chacun des éléments est ensuite modélisé par un agent pour lequel il faut pour définir les connaissances, les capacités fonctionnelles, les comportements et les modes d'interaction qu'il adoptera à l'encontre des autres agents. [**Champion, 03**]

La simulation, à l'instar des observations effectuées dans la réalité, permet d'évaluer le modèle défini et de l'enrichir.

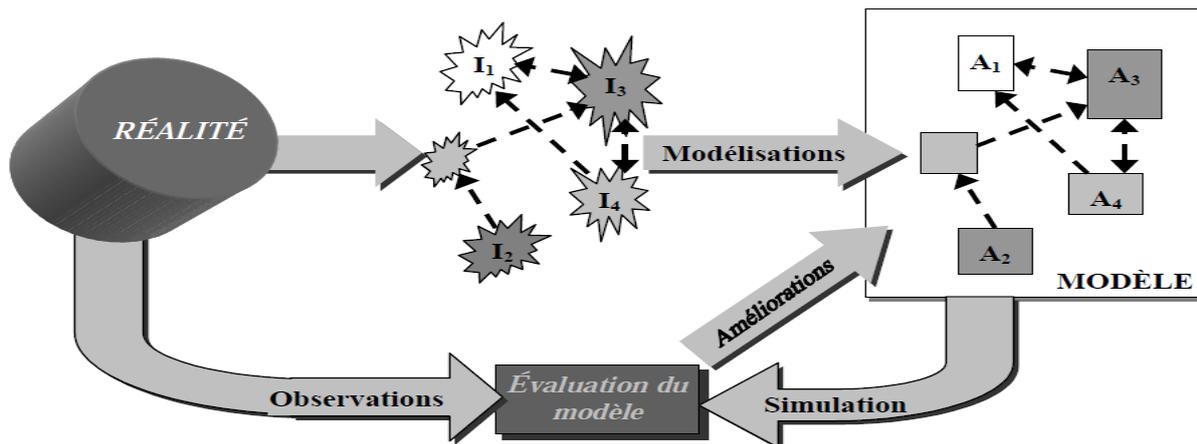


Figure 1.12 : Principes de la simulation multi-agent. [Champion, 03]

La simulation multi-agent est donc une technique qui permet de reproduire des phénomènes difficilement appréhendables de manière analytique. Il s'agit également d'un outil puissant et d'une méthode permettant de découvrir de nouvelles règles d'interaction sociale ou d'organisation.

La simulation multi-agent présente de nombreuses applications et est utilisée, surtout depuis le début des années 1990, dans des domaines extrêmement variés. Il est ainsi possible de citer quelques exemples tels que l'éthologie et plus spécialement la simulation de fourmilières [Drogoul, 93], l'origine et l'évolution du langage humain [Kaplan, 98], la gestion des ressources naturelles les phénomènes urbains, ou encore les organisations humaines.

I.10. Avantage des SMA :

Les SMA-s possèdent les avantages traditionnels de la résolution distribuée et concurrente de problèmes comme [Chaib et al,01]:

- **La modularité** : permet de rendre la programmation plus simple. Elle permet, de plus, aux systèmes multi-agents d'être facilement extensibles, parce qu'il est plus facile d'ajouter de nouveaux agents à un système multi-agent que d'ajouter de nouvelles capacités à un système monolithique.
- **La vitesse** : est principalement due au parallélisme, car plusieurs agents peuvent travailler en même temps pour la résolution d'un problème.
- **La fiabilité** : peut être également atteinte, dans la mesure où le contrôle et les responsabilités étant partagés entre les différents agents, le système peut tolérer la défaillance d'un ou de plusieurs agents. Si une seule entité contrôle tout, alors une seule

défaillance de cette entité fera en sorte que tout le système tombera en panne (dû à la redondance).

Ils héritent aussi des bénéfices envisageables de l'Intelligence Artificielle comme :

- **Le traitement symbolique** (au niveau des connaissances) ;
- **La facilité de maintenance** ;
- **La réutilisation et la portabilité** mais surtout, ils ont l'avantage de faire intervenir des schémas d'interaction sophistiqués.
- **Les types courants d'interaction** incluent la coopération, la coordination et la négociation [**Chaib et al, 01**].

I.11. Problèmes des SMA :

Bien que les SMA offrent de nombreux avantages potentiels, ils doivent aussi relever beaucoup de défis. Voici les problèmes inhérents à la conception et à l'implémentation des SMA, d'après [**Chaib et al, 01**] :

1. Comment formuler, décrire, décomposer, et allouer les problèmes et synthétiser les résultats ?
2. Comment permettre aux agents de communiquer et d'interagir ? Quoi et quand communiquer ?
3. Comment assurer que les agents agissent de manière cohérente i) en prenant leurs décisions ou actions, ii) en gérant les effets non locaux de leurs décisions locales et iii) en évitant les interactions nuisibles ?
4. Comment permettre aux agents individuels de représenter et raisonner sur les actions, plans et connaissances des autres agents afin de se coordonner avec eux ? Comment raisonner sur l'état de leurs processus coordonnés (comme l'initialisation ou la terminaison) ?
5. Comment reconnaître et réconcilier les points de vue disparates et les intentions conflictuelles dans un ensemble d'agents essayant de coordonner leurs actions ?
6. Comment trouver le meilleur compromis entre le traitement local au niveau d'un seul agent et le traitement distribué entre plusieurs agents (traitement distribuée qui induit la communication) ? Plus généralement, comment gérer la répartition des ressources limitées ?
7. Comment éviter ou amoindrir un comportement nuisible du système global, comme les comportements chaotiques ou oscillatoires ?
8. Comment concevoir les plates-formes technologiques et les méthodologies de développement pour les SMA ?

I.12. Champs d'application des systèmes multi-agents

Les SMA sont des systèmes idéaux pour représenter des problèmes possédant de multiples méthodes de résolution, de multiples perspectives et/ou de multiples solveurs.

Un système multi agent possède les avantages de la résolution distribuée et concurrente de problèmes comme la modularité, la vitesse (via le parallélisme), et la fiabilité (due à la redondance). [Sahki, 08]

Ils héritent aussi des bénéfices envisageables d'IA comme le traitement symbolique, la facilité de maintenance, la réutilisation et la portabilité.

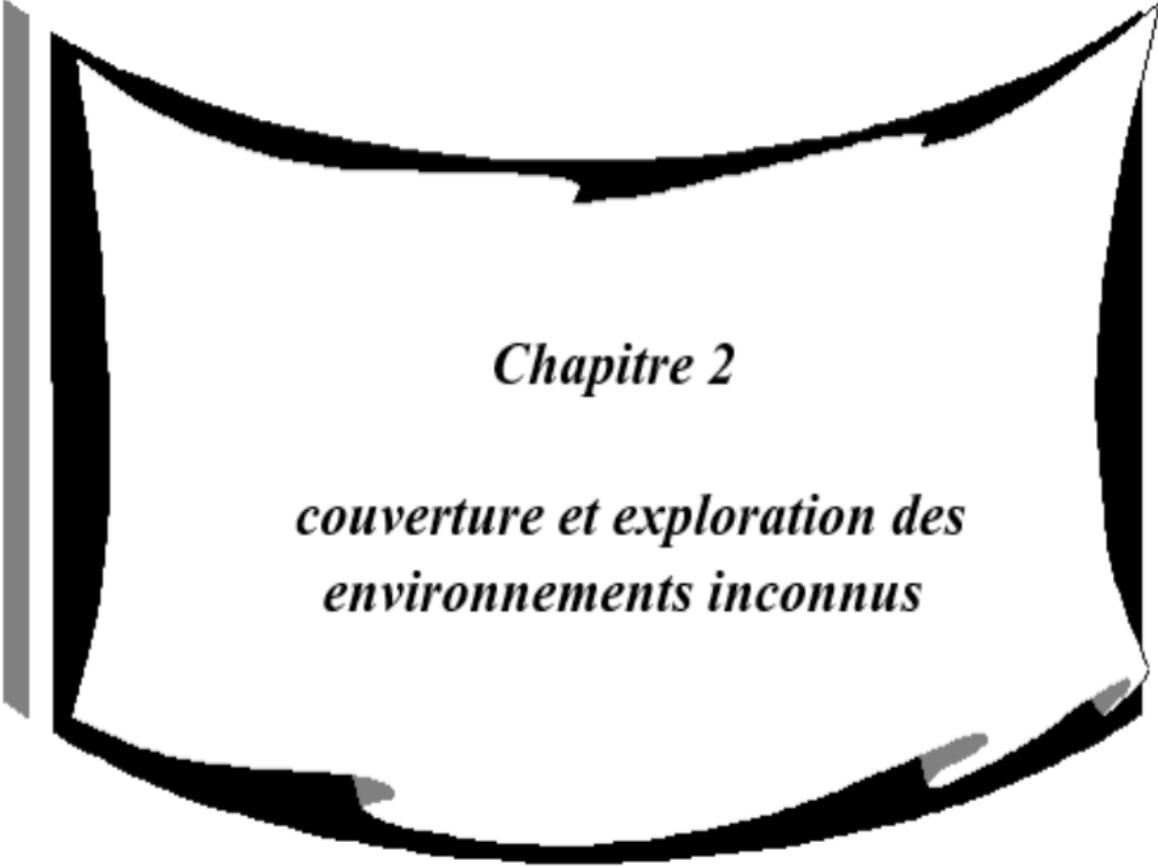
Les domaines d'application de la technologie d'agents sont très vastes, des exemples d'applications sont :

- Assistance du commerce électronique : les agents sont utilisés pour les négociations modérées, en comparant des propositions, etc.,
- Simulation des systèmes décentralisés: agents comme un outil de modélisation (biologie, règlement de la circulation etc.)
- Appareils portatifs et mobiles : les agents dirigent des systèmes mobiles (téléphone cellulaire, PDA, etc.) résolve les problèmes de la connectivité, délégation de tâches de la recherche de l'information, etc.,
- Robots et robot coopération : les agents sont personnifiés dans les robots et réagissent réciproquement dans le vrai monde.
- Interface homme : interface agents (être en face de l'utilisateur), assistants personnels (profil d'utilisateur, apprendre, etc.), des agents qui communiquent et réagissent réciproquement avec les êtres humains (réalité virtuelle, graphique, avatars, etc.,)
- Intégration et inter-opération: agents qui enveloppent des systèmes existants (intégration de l'entreprise, systèmes informatisés, etc.,)

I.13. Conclusion :

Dans ce chapitre, nous avons présenté un état de l'art sur les agents et les systèmes multi-agents. Nous avons commencé par la définition de concept agent, puis nous avons présenté les types des agents existants qui sont classés en général en deux groupes : les agents cognitifs et les agents réactifs, à partir de deux types d'agent nous avons distingué un autre type des agents qui combine entre les deux appelés les agents *hybrides*.

Nous avons indiqué qu'un système où évoluent plusieurs agents est appelé un système multi agents. Ou les agents peuvent interagir entre eux à travers un ensemble d'évènements pendant lesquels les agents sont en relation les uns avec les autres soit directement, soit par le biais de l'environnement. Ensuite nous avons présenté les caractéristiques et les composants de base des SMA-s, en passant par un rappel de la définition d'agents puis la définition d'un environnement jusqu'à l'exploration de relation entre les agents dans un SMA qui est : l'interaction entre eux, ensuite nous avons présenté les principaux comportements d'un SMA à savoir l'auto-organisation, l'émergence et la stigmergie. Comme on a décrit un phénomène naturel qui est « la simulation multi-agent ». Puis nous avons identifié les avantages et les problèmes fondamentaux, Enfin nous avons terminé par les domaines d'application des SMAs.



Chapitre 2

*couverture et exploration des
environnements inconnus*

Introduction

I. Les algorithmes de couverture

I.1. Les algorithmes de couverture pour un seul robot

I.2. Algorithmes de couverture multi-robots

I.3 Les champs de potentiels artificiels

I.4. Comparaison des résultats des algorithmes de couverture mono-robot et multi-robots :

II. L'exploration d'un environnement inconnu

II.1. Définition

II.2. Les algorithmes d'exploration

II.3. Problème d'exploration

III. Conclusion

Introduction

Les robots mobiles jouent un rôle important dans de nombreux domaines, parmi lesquels la police militaire et applications (par exemple, le déminage, les véhicules de patrouille armés, la surveillance, la bombe désarmement), l'industrie (par exemple, transport et de manutention des matériaux), l'espace (par exemple, le plan-exploration monétaire, et l'acquisition des matières sur des planètes lointaines).ils sont utilisés pour les risques des vies humaines et pour l'efficacité assurée pour les machine , la capacité de travail pour des longtemps , et la décharge des travaux fastidieux .

La planification de chemin est une tâche fondamentale dans toutes les opérations de la robotique mobile. Le problème de base pour la planification de chemin d'un robot est de déterminer quel chemin doit être pris par le robot d'un point de départ pour qu'il parvienne à une destination comme dans un environnement connu à l'avance. Une extension de problème planification de chemin est celle de la recherche d'une cible dont sa position est inconnus à l'avance par le robot, ce problème est considéré comme un problème de couverture partielle d'une région ou un problème d'exploration de région. [sarid. 11]

Le problème général de couvrir une superficie de systèmes robotiques simples ou multiples est un problème fondamental en robotique. Elle a des applications dans divers domaines, de missions humanitaires telles que la recherche et le sauvetage et le déminage, aux applications agricoles telles que le semis ou la récolte, à, récemment, d'entretien ménager. Le problème a été largement étudié dans les deux domaines seul robot et des systèmes multi-robots.

Plusieurs approches ont été proposées dans la littérature pour répondre au problème de couverture, on trouvera souvent des travaux utilisant un robot seul et d'autres utilisant plusieurs robots.

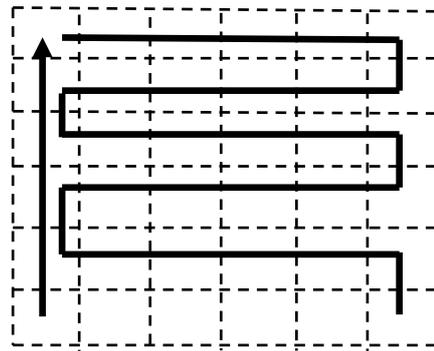
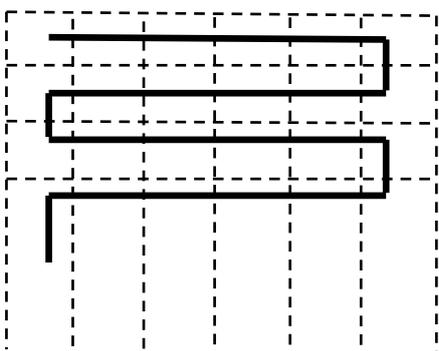
L'exploration d'un environnement par une ou des entités (soit des humains, des véhicules, des robots, ou des agents logiciels) est un sujet ayant capté l'intérêt de nombreux chercheurs, particulièrement dans les domaines des mathématiques et de l'informatique, depuis plusieurs décennies. [Bautin et al, 11]

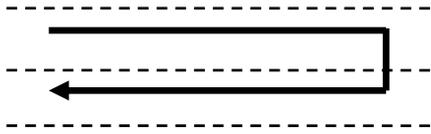
Dans cette partie du chapitre, on va faire un survol sur les plus importants et populaire des algorithmes de couverture qui se trouvent dans la littérature, (le mot agent ou robot sont utilisés interchangeable-ment dans le reste de cette partie).plus une vue générale sur l'exploration des environnements inconnus.

I. Les algorithmes de couverture :

La couverture de l'environnement, une tâche connexe à la patrouille, La couverture consiste simplement à explorer l'ensemble de l'environnement, c'est-à-dire à visiter au moins une fois chaque nœud de l'environnement. L'efficacité d'un algorithme en couverture est simplement mesurée par la durée d'une couverture complète de l'environnement. Cette durée est souvent désignée sous le terme de blanket time. Une couverture optimale consiste, dans un cadre mono-agent, à suivre un chemin hamiltonien. Dans un contexte multi agent, cela revient à diviser l'environnement en un ensemble de chemins hamiltoniens (un par agent) ne s'intersectant pas (c'est-à-dire que chaque agent visitera un nœud inexploré à chaque itération). [Arnaud ,11]

Notons cependant que l'obtention d'une couverture optimale ne garantit pas une patrouille optimale. Si l'on est tenté de considérer intuitivement la patrouille comme une couverture répétée de l'environnement, un simple exemple prouve le contraire. (**Figure 2.1. (a)**) présente une couverture optimale sur un environnement grille. En suivant cette stratégie, une fois que l'agent a effectué la couverture de l'environnement, celui-ci se retrouve dans une situation où sa prochaine décision sera nécessairement sous optimale puisque la cellule de plus grande oisiveté correspondra au point de départ de l'agent (ici en haut à gauche). Une patrouille optimale correspondra cependant à la répétition d'une couverture optimale si ses points de départ et d'arrivée sont adjacents (**Figure 2.1. (b)**). [Arnaud ,11]





(a) une couverture optimale



(b) une patrouille optimale

Figure 2.1: Une couverture optimale ne signifie pas nécessairement une patrouille optimale

I.1. Les algorithmes de couverture pour un seul robot :

Plusieurs méthodes sont trouvées dans la littérature pour la couverture par un seul robot et multi robot. Une méthode de base qui a reçu une attention considérable est la méthode présentée par Gabriely et Rimon, où les auteurs décrivent un Spanning algorithme de couverture d'arbre, connu comme l'algorithme SC. Dans cette méthode, trouver un cycle hamiltonien couvrant terrain qui répond à certaines hypothèses. En particulier, ils ont supposé que le robot est équipé d'un outil de forme carrée de la taille D , d'où la zone a été divisée en N cellules de taille D placés sur une grille. La grille était alors en gros de sorte que chaque nouvelle cellule est de taille $2D \times 2D$, et un arbre couvrant été construit selon cette nouvelle grille. Après un tel arbre a été construit, le robot suit l'arbre autour, créant un cycle hamiltonien pour visiter toutes les cellules de la grille d'origine. [Noam et al, 05]

De nombreux algorithmes de couverture utilisent des graphes pour représenter l'environnement. L'algorithme de base de couverture sur un graphe est le DFS (depth-first-search), qui explore en profondeur dans le graphe chaque fois que possible, et retourne en arrière sinon. Cet algorithme tente d'abord d'atteindre la solution le plus vite possible en explorant immédiatement les successeurs de tout noeud généré, alors que la seconde étend l'arbre en générant les nœuds couchent par couche. A chaque étape, l'algorithme met à jour la file des nœuds non explorés. C'est toujours le premier nœud de la file qui est étendu. Les nœuds résultants sont ajoutés en tête de la file de sorte qu'ils soient explorés en premier dans les étapes ultérieures.

Les avantages de la stratégie profondeur d'abord sont la simplicité de son implémentation et le fait que l'algorithme ne requiert que très peu de mémoire. L'un des problèmes majeurs de cet algorithme se pose lorsque le graphe contient des cycles. Cette possibilité est donnée lorsque les opérateurs permettent des opérations réversibles.

- **Spanning trees (ST) :**

La méthode proposée par [Gabriely et al, 01] est une heuristique reposant sur l'utilisation de spanning trees et permettant de construire des cycles hamiltoniens. En théorie des graphes, les spanning trees permettent la représentation de graphes par des arbres de taille finie. Cette représentation en arbre permet de représenter l'ensemble des nœuds du graphe sans risquer de rencontrer d'éventuels boucles ou cycles.

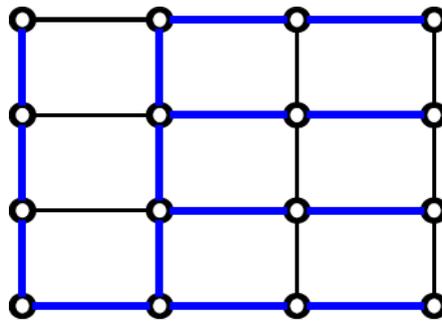


Figure 2.2 : Un spanning tree (arcs bleus) d'un graphe grille

La méthode proposée repose donc sur l'utilisation de tels arbres. Considérant un environnement grille avec des cellules de côté de longueur n , les auteurs lui superposent une seconde grille, de granularité plus importante, avec des cellules de côté de longueur $2n$. Ainsi, à chaque cellule du graphe grossier sont associées 4 cellules du graphe original. Un algorithme de construction de spanning tree est ensuite exécuté sur le graphe grossier. L'agent n'a alors plus qu'à suivre le spanning tree (chemin fléché, Figure 2.2) pour effectuer une patrouille optimale de l'environnement.

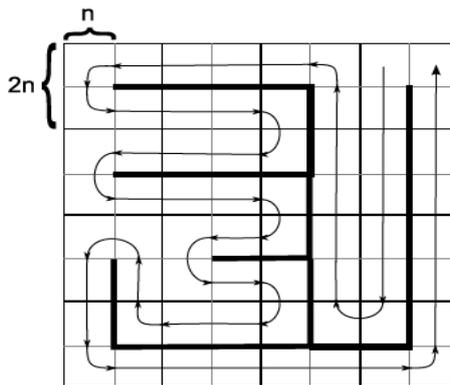


Figure 2.3 : Un cycle hamiltonien réalisé à partir d'un spanning tree sur un environnement grille de 8x8 nœuds.

Si cette méthode permet de planifier une couverture optimale très rapidement (en temps linéaire $O(N)$, N étant le nombre de cellules de l'environnement), elle reste cependant assez limitée et pose des contraintes fortes sur l'environnement :

_ Le graphe est contraint à être une grille à voisinage de Von Neumann (chaque nœud à au plus 4 voisins) dont les côtés sont de longueurs paires.

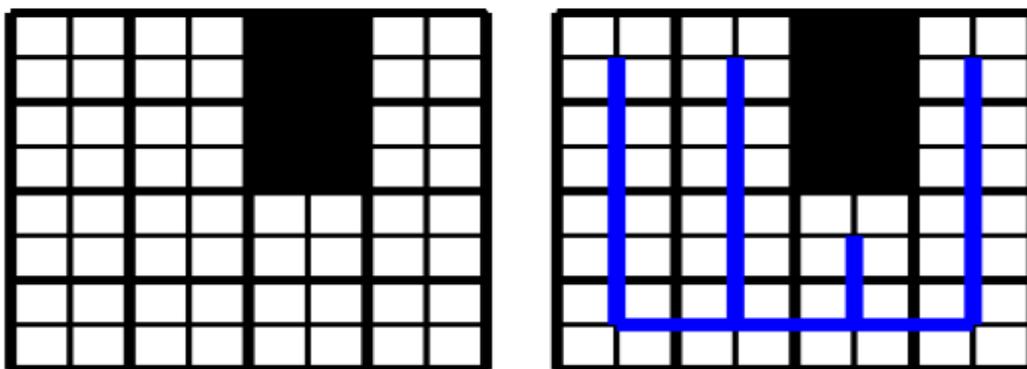
_ Les obstacles présents dans l'environnement peuvent rendre cette méthode inopérante si leur densité est importante.

- **Algorithme Spanning Tree Couvrant (STC) :**

La structure du Spanning Tree a un rôle essentiel dans le temps de la couverture obtenue par des algorithmes qui utilisent l'arbre comme base pour la couverture. N'importe quel algorithme de couverture, ne peut pas atteindre le temps de couverture bas que l'on peut obtenir en utilisant un arbre différent. Une Spanning Tree, ce qui en soi obtient le temps d'une couverture optimale, il n'existe pas nécessairement, par conséquent, le temps d'une couverture optimale théorique peut rester inaccessible dans certains cas.

L'environnement que les robots fonctionnent en est représentée comme une grille 2D de grandes cellules carrées, qui peut être soit complètement bloqué ou totalement débloqué. Chacune de ces grandes cellules ont composé de quatre petites cellules. Tous les robots sont de la taille d'une petite cellule, et peut être situé dans une de ces cellules à tout moment, si elle est débloquée.

Un robot qui se trouve dans une cellule non bloqué peut se déplacer à une cellule adjacente, s'il est débloqué (ce qui signifie que le robot peut se déplacer vers le haut, bas, gauche et droit, mais pas dans les mouvements en diagonale). Une telle démarche suppose de prendre le temps de l'unité (le temps est le même pour tous les robots). Il est possible que plus d'un robot puisse occuper la même cellule à la fois.



(A) une représentation de l'environnement

(b) une représentation de Spanning Tree

Figure 2.4 : Exemple de l'algorithme STC

Gabriely et Rimon ont introduit trois versions de l'algorithme Spanning Tree covering (STC), qui assurent une couverture optimale : les algorithmes off-line, on-line, et Ant-like STC.

- **Algorithme STC off-line :**

En premier temps, le robot est équipé d'un outil couvrant rectangulaire de taille D et peut se déplacer seulement dans les quatre directions orthogonales à côté de l'outil. Un arbre recouvrant est construit en utilisant DFS sur une grille sans obstacles d'une taille $2D$. Finalement, chaque cellule $2D$ est subdivisé en quatre cellules de taille D . et le robot fait le tour de cette arbre jusqu'à ce qu'il atteigne le point de départ, ce qui implique une couverture complète. [Sarid.2011]

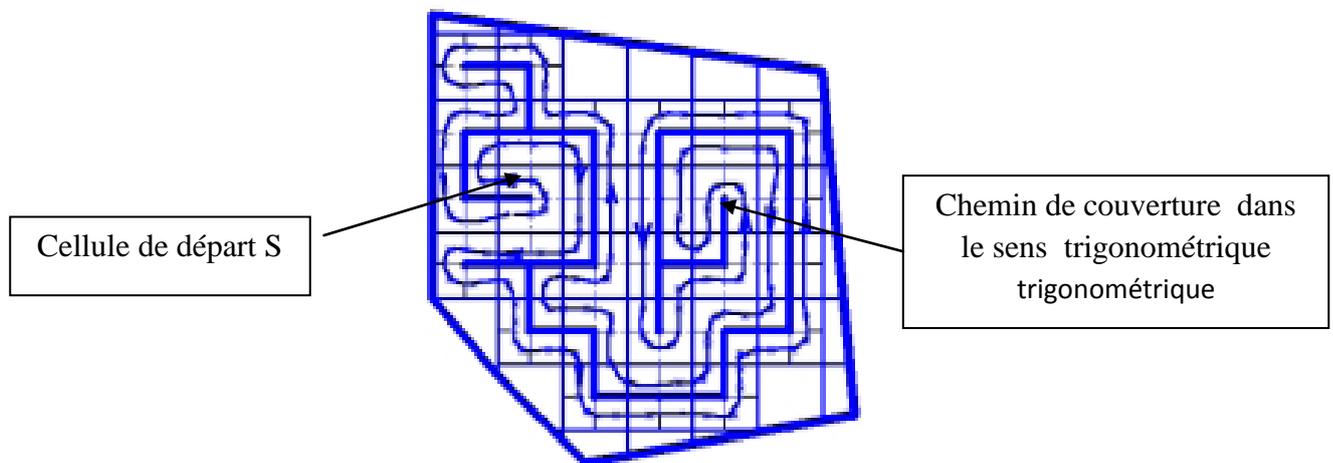


Figure 2.5 Un exemple d'exécution de l'algorithme STC off ligne.[Gabriely et al, 1999]

- **Algorithme STC On-Line:**

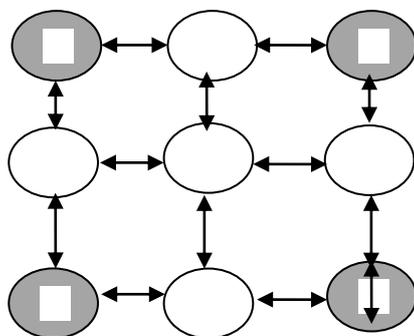
Dans la version en ligne de STC le robot n'a pas de connaissance préalable de l'environnement, sauf que les obstacles sont fixes. Au contraire, le robot doit utiliser ses capteurs embarqués pour détecter les obstacles et de planifier son chemin couvrant en conséquence. [Gabriely et al,99] le robot a capteurs de position et d'orientation, ce qui lui permet de reconnaître les cellules $2D$ de taille comprenant la zone de travail. Et aussi que le robot a un détecteur de gamme, capable d'identifier les obstacles dans les quatre cellules voisines cellule actuelle .Les questions pratiquement importants de sélection des capteurs, des erreurs de mesure du capteur, et la fusion de capteurs ne sont pas considérés. Au contraire, ils ont supposé que les capteurs sont idéales, et qu'ils fournissent des lectures parfaites. Dans l'algorithme STC en ligne, le robot construit progressivement un arbre de recouvrement de la

grille représentant la zone de travail. Lors de la construction d'arbre couvrant, le robot subdivise chaque cellule qu'il rencontre en quatre sous-cellules identiques de taille D , chacune étant identique à la taille de robot. Cet outil de couverture suit le chemin de la partie de cellule qui contourne l'arbre couvrant progressivement, jusqu'à ce que l'ensemble de la grille de région est couverte.

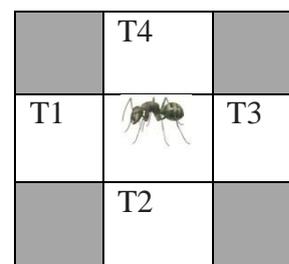
- **ANT-like STC Algorithm:**

Chang et al ont utilisé une fourmi robot mobile comme un modèle de robot simplifié d'une vraie fourmi. Dans ce modèle, les robots de fourmi et de l'environnement ont plusieurs caractéristiques importantes:

- Une fourmi robot vit dans le domaine discret.
- La carte et les obstacles sont constitués de carreaux qui peuvent être considéré comme sommets dans un graphe non orienté G
- L'environnement est fini et limité.
- Dans chaque cycle de temps, les robots-fourmis n'effectuer aucune action à des instants identiques. Ils se déplacent de façon aléatoire dans la période de temps afin d'éviter le conflit d'occuper la même case dans le même laps de temps.
- Une fourmi ne peut se déplacer à quatre tuiles voisines ou sommets. En cas d'égalité, la fourmi fera une spiralmovement décision.
- Chaque mouvement coûte une unité de temps et une unité de énergie.
- La communication entre les fourmis sont basées sur l' phéromone (marques) a laissé sur les carreaux.
- Une fourmi a une mémoire limitée. [Jacky et al]



(a)



(b)

Figure 2.6 : La fourmi peut se déplacer à carreaux ou des sommets, T1, T2, T3 et T4. (a) Le cercle est la plage du capteur efficace de la fourmi. (b) La forme équivalente dans un graphe non orienté. [Jacky et al]

La version Ant-Like STC est aussi un algorithme on-line, elle utilise les mêmes capteurs décrits ci-dessus. Cependant, maintenant que le robot a la capacité de laisser des marques dans les sous-cellules qu'elle couvre, en utilisant la couleur, l'odeur, la chaleur, ou des marqueurs de galets. Les Auteurs ont supposés en outre la disponibilité d'un dispositif de détection, capable d'inspecter les marques sous-cellules dans la cellule actuelle et ses quatre voisins immédiats.

Utilisant cette dispositif de détection, le robot identifie une cellule voisine en tant que nouveau lorsque ses quatre sous-cellules sont toutes non marqué. Tout comme l'algorithme STC on-line, ici aussi, le robot utilise DFS pour construire progressivement un arbre couvrant de la région. Cependant, plutôt que de stocker le couvrant arbre dans sa mémoire, le robot utilise les marques sous-cellules pour identifier des cellules-mères le long de spanning tree comme suit. Par construction, le Robot traite et marque les sous-cellules d'une case donnée dans le sens trigonométrique. Par conséquent, tant que les quatre sous-cellules de la cellule actuelle ne sont pas tous marqués, le robot n'a qu'à analyser les sous-cellules dans l'ordre anti-horaire et d'identifier la transition entre sous-cellules non marqué et marqué. Les deux sous-cellules qui bordent cette transition ont un bord extérieur en commun. cet avantage est nécessairement le bord de la frontière entre la cellule active et sa cellule-mère dans le spanning tree. En outre, le robot n'a pas besoin d'identifier une cellule mère d'une cellule dont les quatre sous-cellules sont déjà marqué, car une fois que le robot couvre les quatre sous-cellules d'une cellule donnée, il ne revient jamais à cette cellule à nouveau. [Gabriely ,99].

- **Algorithme Spiral STC :**

Les auteurs [Gabriely et al,02] démarrent par une version préliminaire de l'algorithme appelé 2D-Spiral-STC. L'algorithme préliminaire ne couvre que les cellules de taille 2D qui sont complètement sans obstacles. Ces cellules sont couvertes par un chemin optimal induite par un arbre couvrant dont les noeuds sont des cellules de taille 2D libres. L'algorithme complet couvre grilles généraux en traitant les cellules de taille 2D qui sont partiellement occupés par des obstacles comme des nœuds spéciaux du Spanning Tree qui effectué une couverture répétitive.

Le spiral STC améliore l'algorithme préliminaire par une couverture des cellules 2D partiellement occupée. Ces derniers sont définies comme les cellules qui contiennent au moins une sous-cellule (D-size) sans obstacle. Cet algorithme utilise la structure de graphe augmenté suivante : les nœuds sont les centres des cellules libres et occupées, les bords du graphe connectent les centres des cellules adjacentes qui contiennent des sous-cellules sans obstacles avec des frontières communes. Cette construction donne naissance à deux types de bords :

Le premier type, appelé bords recto-verso, ne possède qu'une sous-cellule libre sur les deux côtés (Figure 2.7 (a)). L'autre type, appelé bords simple face, possède au moins une sous-cellule occupée de chaque côté (Figure 2.7 (b)). Spiral-STC construit progressivement un arbre couvrant du graphe augmenté, le tour d'un bord simple face aboutit à une couverture répétitive de certaines sous-cellules associées à ce bord. Voici un exemple représenté par une telle cellule avec deux nœuds, chacun ayant des bords de cellules adjacentes à partir de laquelle une sous-cellule libre peut être consultée (Figure I.14 (c)). [Gabriely et al ,02]

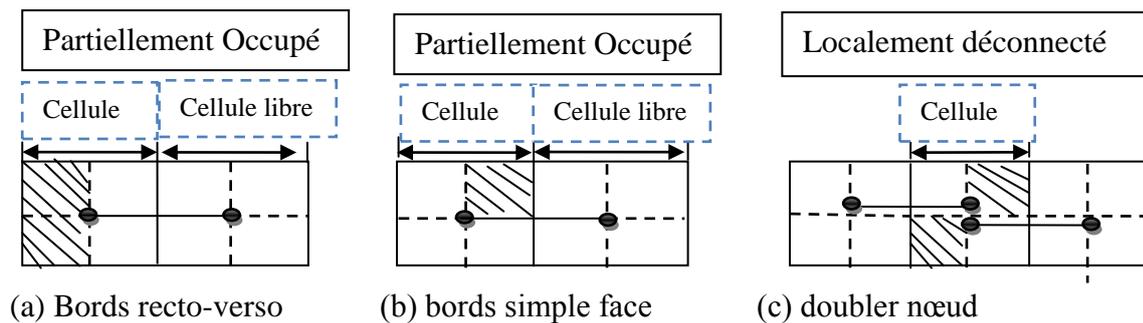


Figure 2.7 : (a) une seule face, (b) à double face bord, (c) doublement de nœud à une cellule déconnectée.

L'algorithme complet Spiral-STC a la même structure que l'algorithme préliminaire. Par conséquent, ils ont décrit seulement la fonction récursive, appelé STC2 (w, x), qui est modifiée afin de tenir compte des deux types de bords qui se produisent dans la grille graphique augmentée. [Gabriely et al ,02].

I.2. Algorithmes de couverture multi-robots :

Tandis que les algorithmes de protection mono-robots ont reçu beaucoup d'attention, il existe actuellement de nombreux algorithmes de moins pour le problème de couverture multi-robots. La plupart des algorithmes de couverture multi-robots sont des robots qui interagissent et

prévoient que localement, souvent appelés robots de fourmis. Naturellement, la planification globale peut conduire à un significativement petit moment de couverture, car il permet aux robots de coordonner leurs trajectoires beaucoup mieux.

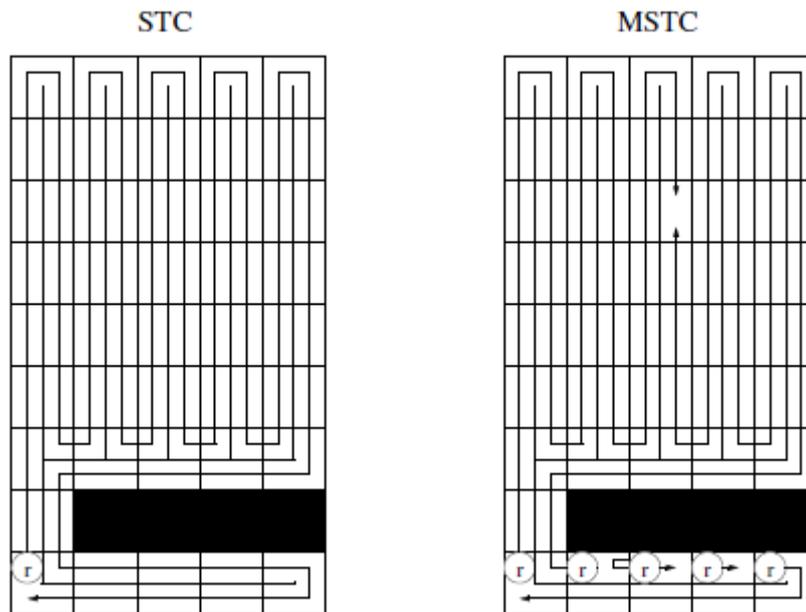
- **Algorithme MSTC :**

La généralisation de l'algorithme STC pour un système multi-robots a été introduite par Hazon et Kaminka. [Noam et al, 05] Ils ont présenté plusieurs algorithmes pour une couverture multi-robots d'un terrain, qui garantissent une couverture robuste, temps complet et efficace.

Ils ont dérivé deux versions de l'algorithme MSTC: MSTC de non backtracking et MSTC backtracking, ci-après dénommé NB-MSTC et B-MSTC, respectivement. Dans l'algorithme MSTC NB les robots se déplacent simplement dans le sens antihoraire ou long du chemin de Spanning Tree jusqu'à la position initiale du robot suivante si aucune panne ne se produit, ou de prendre sur le trajet de la couverture du robot consécutive contraire. Dans le B-MSTC les robots peuvent revenir en arrière sur certaines parties de leur trajectoire de couverture, c'est à dire, ils peuvent aller à la fois dans le sens horaire et antihoraire. Ils ont montré que si le robot marche arrière, le pire des cas effectue jusqu'à deux fois plus rapide que dans le cas de non backtracking, malgré la redondance. [Hazon et al,]

[Sarid et al ,11] Ils ont explorés un problème de planification de mouvement en ligne pour un groupe de robots à la recherche d'une cible dont la position inconnue.

Récemment, la STC a été généralisé à Multi-Robot couverture Spanning Tree (MSTC), un polynôme-temps multi-robots heuristique de couverture. MSTC première calcule le même arbre couvrant que STC, et considère que la tournée qui fait le tour de l'arbre couvrant. Chaque robot suit le segment de la tournée dans le sens horaire devant lui, à une exception près: Pour améliorer le temps de couverture, le segment le plus long est divisé à parts égales entre les deux robots adjacents. Quelques petits ajustements, alors assurez-vous que MSTC réduit le temps de couverture du STC par un facteur d'au moins 2 (ou $3/2$) pour $k \geq 3$ robots (ou deux robots, respectivement). Chaque petite cellule est visitée par un seul robot, donc il n'y a jamais eu de collisions ou des blocages.



• **Figure 2.8 : exemple de MSTC en fonctionnement.**

• **Algorithme MRSAM :**

L'algorithme MRSAM lance plusieurs robots à partir d'un point de départ commun S et attribue chaque robot j à un disque à la recherche de la cible T, tous les disques sont concentriques, et S est leur centre.

Dans [Sarid et al ,11], Ils ont introduit un nouvel algorithme de navigation en ligne, appelé MRSAM (le Multi-Robot Search Time multiplication), qui est basée sur le doublement de la superficie de l'algorithme SAD1 (Navigation à une cible inconnue). L'idée principale de SAD1 est comme suit : Le robot sélectionne un disque initial de rayon R_0 centré à S, et recherche dans la partie de ce disque accessible à partir de S. Si la cible est détectée l'algorithme se termine. Sinon, le robot répète le processus sur les disques de rayons $2i = 2R_0$ pour $i = 1; 2;$ jusqu'à ce que la cible est détectée, ou toute la région accessible à partir de S est explorée sans trouver T (Figure 2.9). Le processus de recherche dans chaque disque est comme suit. Le robot impose une discrétisation en ligne de la région continue en une grille de cellules de taille D. La grille se compose uniquement de cellules libres et est entourée par des cellules partiellement occupés. Le robot exécute un tour de la zone de couverture de série sur la grille de cellules libres, tout en balayant chaque nouvelle cellule de la cible. A l'entrée d'une nouvelle cellule, le robot analyse en plus les cellules voisins partiellement occupés pour T. Si la discrétisation préserve la connectivité de la région accessible (cette hypothèse peut être assouplie par un algorithme plus complexe qui surveille la rupture de la connectivité locale), il

est clair que toutes les cellules libres et partiellement occupé dans la région accessible à partir de S sont finalement inspectées par le robot.

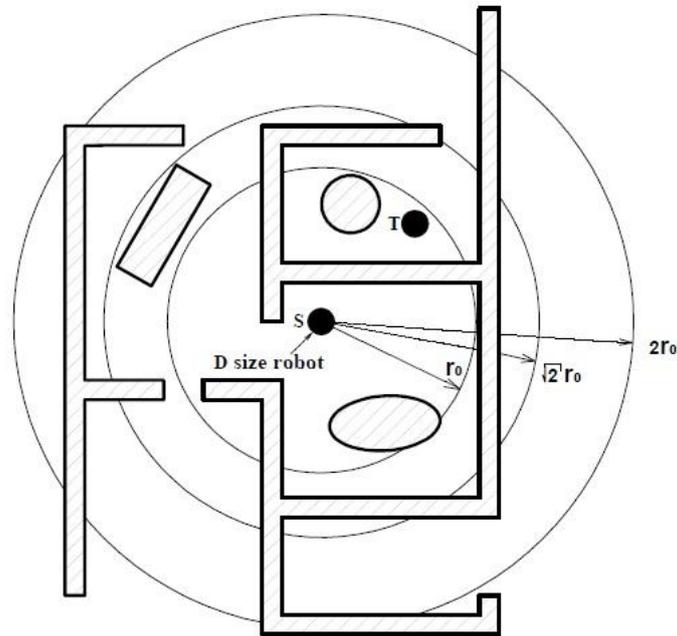


Figure 2.9: La région du disque recherché par SAD1 doublé à chaque étape.

Il nécessite de la mémoire linéaire et a un rendement concurrentiel quadratique. En outre, il est démontré que de façon générale tout algorithme de navigation en ligne doit avoir au moins une performance compétitive quadratique. L'algorithme MRSAM atteint des limites inférieures donc il a une compétitivité optimale. MRSAM lance un groupe de robots à partir d'un point de départ commun pour effectuer la recherche de la cible. Chaque robot est du type D et a un système de positionnement qui est supposé être idéal, un tactile ou un capteur de vision restreint qui détecte des obstacles dans l'environnement et un capteur de reconnaissance de cible, qui peut être n'importe quel capteur selon le type de la cible qu'il est l'objet.

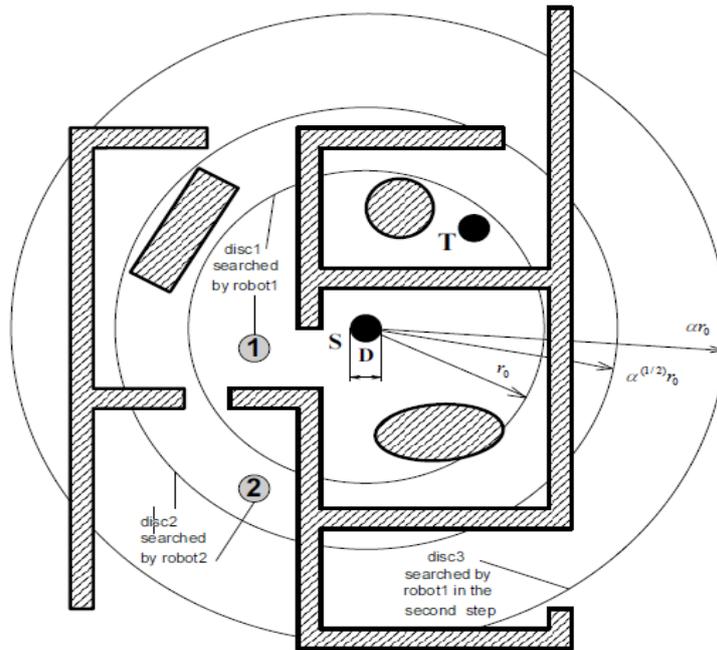


Figure 2.10 : Un groupe de deux robots lancés par MRSAM à la recherche de la cible

Chaque robot est attribué à un disque, pour rechercher la cible, chaque disque a une taille différente (voir **Figure 2.10**).

Le premier robot est affecté à un disque avec surface S_0 et chaque robot conséquent est affecté à un disque de surface supérieure à la surface du robot précédente par un facteur de multiplication $\alpha > 1$, qui ne dépend que de n , le nombre de robots allouée à la tâche. Chaque robot cherche la cible dans son disque jusqu'à ce qu'il trouve la cible ou jusqu'à ce qu'il finit la couverture de tous les secteurs liés au disque, dans ce dernier cas, le robot est affecté à poursuivre à la recherche dans un autre disque libre.

les auteurs (Shahar Sarid et Amir Shapir) font quelques remarques informelles sur l'algorithme. Tout d'abord, lors de la section d'initialisation, après l'obtention des valeurs de n et r_0 , chaque robot peut calculer sa future Disques de recherche et les rayons correspondants, ce qui signifie que les robots n'ont pas besoin de communiquer les uns avec les autres, à l'exception d'un signal d'arrêt lorsque la cible est trouvée. Cela implique une approche décentralisée avec pas ou peu de communication. Deuxièmement, un robot qui a terminé la recherche d'un disque procédera immédiatement à l'autre disque qui lui est assignée. La visite de la couverture peut être exécuté avec un algorithme DFS trivial en utilisant la mémoire linéaire. Performance moyenne de l'algorithme peut être améliorée si chaque robot couvre à chaque étape seules les cellules qui se trouvent dans le cycle ajoutés au disque précédent.

Quatrièmement, si la cible est inaccessible à partir de S , l'algorithme ne s'arrête que lorsqu'il est complètement couvert la composante connexe de l'environnement contenant S . [Sarid et Shapir,].

Multi-Robot Forest Coverage (MFC) :

Les auteurs [Zheng et al,] ont décrits la couverture forestière Multi-Robot (MFC). Il est basé sur l'algorithme de Even et al. qui donne un four approximation pour le problème de trouver une couverture d'arbre avec des racines donnés, en minimisant le poids de l'arbre le plus lourd.

MFC fonctionne sur le graphe dont les sommets sont les grandes cellules, et dont les bords se connectent aux grandes cellules bloquées adjacentes. Si les robots commencent dans une grande cellule, puis MFC fait r copies identiques de ce sommet. MFC trouve un couvert végétal racines de ce graphique en temps polynomial, où les racines sont les sommets qui contiennent des robots. (Le graphique est autorisé à être déconnecté, aussi longtemps que chacun de ses composants contient au moins un robot.) Chaque robot contourne alors son arbre.

En conséquence, la couverture forestière donne un quatre-approximation pour la couverture forestière. Il fonctionne comme suit:

- 1) Retirer toutes les arêtes avec des coûts de pointe supérieures à celles B .
- 2) Contracter tous les racines dans un seul sommet, Trouver un arbre recouvrant minimal pour le graphe résultant, puis à nouveau sur la uncontract seul sommet,diviser le spanning tree dans les arbres $|R|$.
- 3) Décomposer chaque arbre en sous-arbres qui peuvent partager sommets, mais pas les bords. Le poids de chaque sous-arbre est dans l'intervalle $[B; 2B]$, à l'exception possible d'une sous-arborescence qui contient les restes de la racine de l'arbre, et dont le poids est inférieur à B .
- 4) Trouvez un couplage maximum de tous les sous-arbres non restes des racines, sous la contrainte que non seulement les restes de sous-arbre peut être associé à une racine si le non-restes arbre arborescence et restes de la racine (root ou lui-même) sont à distance au plus B . Si certains sous-arbres non restes ne peuvent être jumelés, c'est la preuve qu'aucune couverture arbre enraciné de poids au plus B existe.

- 5) Pour chaque racine, retourner un arbre constitué de la racine, les restes de sous-arborescence de la racine (le cas échéant) de poids au plus B , le seul non-restes arborescence adaptée à la racine (le cas échéant) de poids au plus $2B$, et un chemin coût minime de poids au plus B de la non-restes-arbre à la restes arborescence (ou root). Le poids de chaque arbre est dans la plupart des $4B$, résultant en un couvercle d'arbre ancré de poids d'au plus $4B$. [Xiaoming et al,]

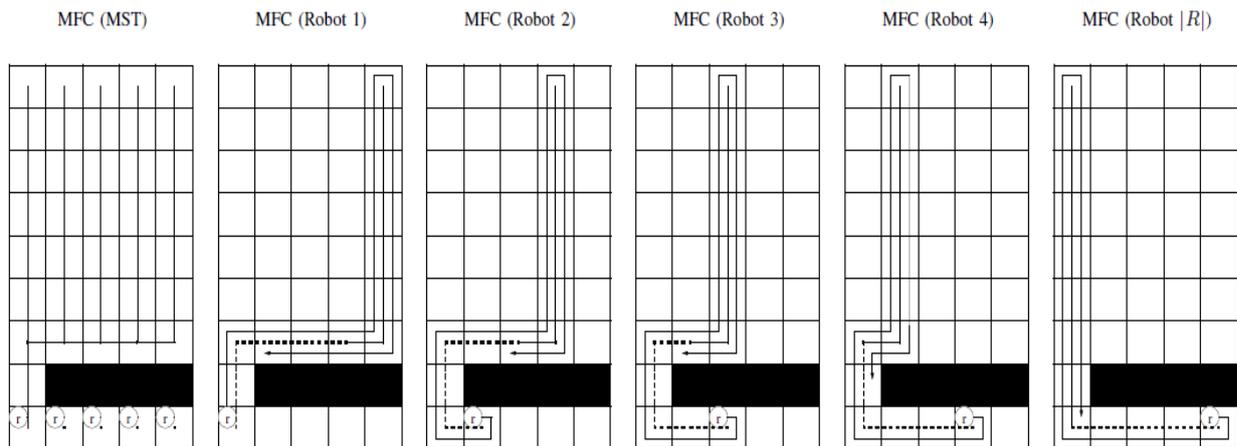


Figure 2.11 : exemple de l'algorithme MFC

I.3 Les champs de potentiels artificiels :

Les champs de potentiel [Khatib ,86] est une méthode assez originale qui assimile le robot à une particule soumise à un champ de forces répulsives et attractives. Un obstacle génère un champ de potentiel répulsif tandis que l'objectif à atteindre génère un champ de potentiel attractif. L'algorithme calcul donc un vecteur résultant qui indiquera au robot comment effectuer son déplacement.

Cet algorithme est totalement réactif et peut donc être très facilement implémenté en temps réel. Cependant, comme l'environnement est très peu souvent totalement convexe, cette méthode plonge facilement le robot dans des minimas locaux. De plus, les problèmes d'oscillation peuvent, dans certain cas, être constatés.

I.4. Comparaison des résultats des algorithmes de couverture mono-robot et multi-robots :

I.4.1. MFC & STC & MSTC :

-S'il y a un seul robot, MFC réduit à STC et réduit ainsi le temps de couverture. S'il y a plus d'un robot, rappeler que MSTC réduit le temps de couverture du STC par un facteur d'au moins 2 (ou 3/2) pour $k \geq 3$ robots (ou deux robots, respectivement). MFC ne peut pas prendre une telle garantie solide de pire des cas sur la façon dont son temps de couverture est bon et concernant le plus petit temps de couverture d'un seul robot.

-Le temps de couverture du MFC ne peut pas être plus mauvais que celle de STC parce que MFC rend chaque robot un arbre qui peut être étendue à un arbre couvrant. D'autre part, la figure 4 montre un exemple où le temps de couverture du MFC est égale à la durée de couverture du STC (où, dans le cas du STC, le deuxième robot ne bouge pas), peu importe combien de temps le corridor est, même si le temps de couverture du MSTC est seulement la moitié du temps de couverture du STC. [Zheng et al,]

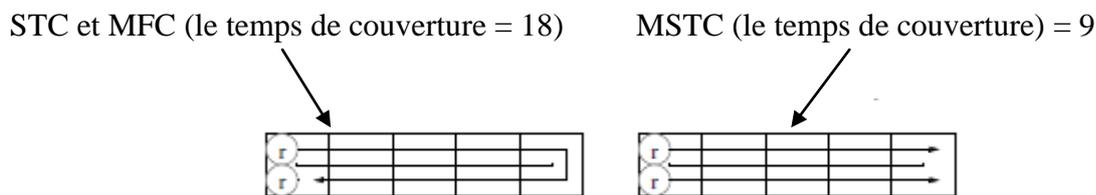


Figure 2.12. MFC et MSTC contre STC

-Les résultats expérimentaux des auteurs [Xiaoming al , 05] montrent que le temps de couverture du MFC est plus petit que celui de Multi-Robot couverture Spanning Tree (MSTC) et proche de l'optimum dans tous les scénarios testés.

I.4.2. Off-line STC & On-line STC & Ant-like STC :

-Les trois algorithmes STC: Off-line STC, On-line STC, et Ant-like STC, ne couvrent pas de manière répétitive tout moment la zone de couverture.

-Les trois algorithmes STC couvrent chaque cellule qui est accessible à partir de la cellule de départ S.

-Les trois algorithmes STC couvrent complètement une grille connectée représentant une zone de couverture continue. De plus, le revêtement est optimal en ce sens qu'il n'y a pas de couverture répétitive d'un point quelconque de la zone continue sous-jacente à la grille.

-L'On-line STC maintient une structure de données de l'arbre couvrant construit progressivement, tandis que l'Ant-like STC utilise des marques de sous-cellule pour identifier localement l'arbre couvrant.

II. L'exploration d'un environnement inconnu

L'exploration d'un environnement par un groupe d'entités est un problème algorithmique fondamental ayant des applications directes dans plusieurs domaines, soient-ils physiques ou virtuels. En effet, on peut penser à l'exploration d'un labyrinthe par plusieurs personnes à la recherche d'un trésor, l'exploration d'un environnement inhabitable (tel qu'une planète lointaine) par un groupe de robots, la planification des meilleures routes pour la livraison de biens par une flotte de véhicules, l'inspection d'un bâtiment par un groupe de pompiers à la recherche de victimes ou l'exploration de nœuds dans un réseau informatique.

Nous cherchons une stratégie permettant aux robots de visiter toutes les zones de l'environnement le plus rapidement possible. Dans ce contexte, l'utilisation de plusieurs robots est avantageuse mais le gain est conditionné par le niveau de coopération entre les robots.

II.1. Définition

Un environnement inconnu est défini par le fait que l'information nécessaire à l'exploration à coût optimal n'est pas disponible lors de l'exécution de l'algorithme d'exploration. Par exemple, dans l'étude faite par Rao et al. [**Rao et al**], un terrain inconnu est défini par le manque d'information sur le modèle du terrain (les obstacles, les chemins, la carte du terrain, etc.). D'autres auteurs utilisent cette même classification. De même, le manque de connaissances dans un environnement non géométrique, tel un graphe, peut être défini par le manque d'information sur l'identification des sommets ou des arêtes (p. ex. [2, 6, 7, 8, 17, 21, 18]).

Dans son sens le plus large, l'exploration se définit par le fait de percevoir (soit physiquement, visuellement, virtuellement, ou avec un capteur quelconque dans le cas d'un robot) tous les éléments de l'environnement dans un but donné. Le but de l'exploration peut être, par exemple : la découverte d'un objet, l'exécution d'une tâche à un endroit dans l'environnement, la cartographie de l'environnement, etc. [**Frédéric, 13**]

II.2. Les algorithmes d'exploration

Dans cette section nous présentons les principaux types d'approches existantes pour l'assignation de frontières aux robots pour l'exploration d'environnements inconnus.

Les premières méthodes d'exploration étaient basées sur une navigation aléatoire [Lopez et al, 97] ou le suivi de mur [Tom, 97]. En mono-robot, comme en multirobot, une technique admise comme efficace est l'exploration successive des frontières créées entre les zones non-explorées et les zones explorées accessibles [Brian, 97]. Les robots se déplacent vers les frontières et collectent ainsi de nouvelles informations sur les zones inexplorées de l'environnement.

II.2.1. Frontière la plus proche

Yamauchi est le premier à avoir développé une stratégie multi-robot d'exploration de frontière [Brian, 97]. Les robots partagent les informations locales récoltées afin que chacun d'entre eux produise une carte similaire fournissant une liste de frontières similaires. Chaque robot se dirige vers la frontière la plus proche de lui, effectue une observation et en diffuse (broadcast) Aucune communication n'est nécessaire pour coordonner les robots. C'est un système asynchrone, distribué et robuste aux pannes de robots.

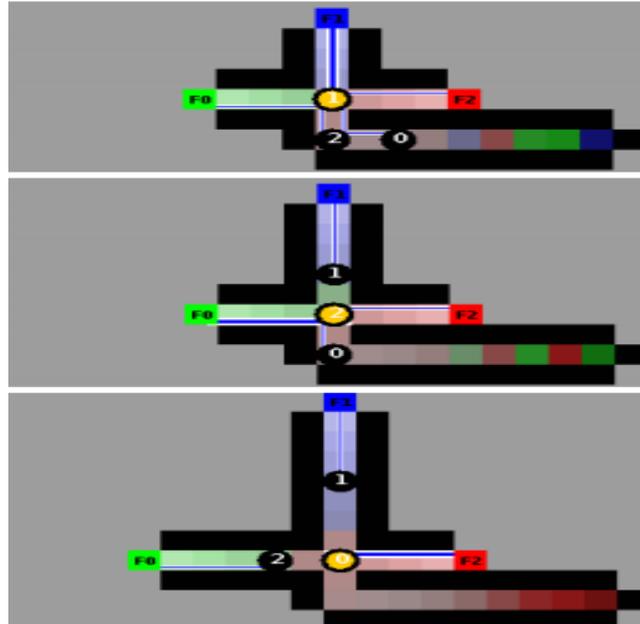


Figure 2.13. Coordination implicite avec l'algorithme frontière la plus proche [Bautin et al, 11]

Lors de son application, une coopération implicite s'effectue (voir la figure 2) mais elle est limitée car les robots peuvent choisir les mêmes frontières ne tirant pas avantage de leur nombre.

Description de l'image (du haut vers le bas) : quand le premier robot R1 atteint l'intersection des couloirs il découvre 3 frontières équidistantes et choisit aléatoirement

la frontière F1, quand R2 atteint l'intersection, il reste 2 frontières plus proches équidistantes et choisit aléatoirement F0 entre F0 et F2, quand le 3ième robot R0 arrive sur l'intersection une seule frontière est plus proche de lui et il choisit donc F2.

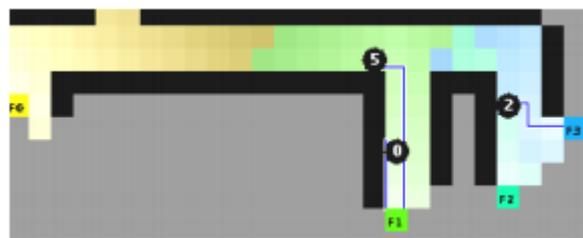


Figure 2.14. Résultat de l'assignation avec l'algorithme Frontière la plus proche [Bautin et al, 11]

II.2.2. Glouton

L'algorithme Glouton tente d'optimiser le coût total de l'exploration en fixant à chaque itération la paire robot-frontière ayant le coût minimum. Comme il itère sur les robots .Cet algorithme est en général appliqué de manière centralisée mais si chaque robot dispose de la matrice de coût, son exécution décentralisée est possible (chaque agent exécute l'algorithme jusqu'à obtenir son assignation), [Bautin et al, 11]

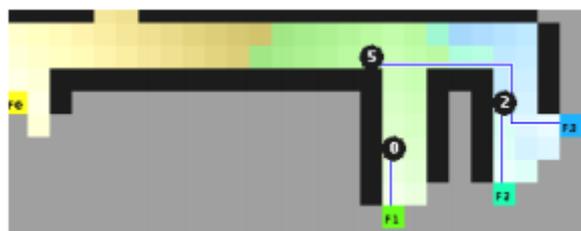


Figure 2.15. Illustration de l'allocation de frontière avec l'algorithme glouton [Bautin et al, 11]

II.3. Problème d'exploration

L'exploration d'un environnement inconnu est un problème fondamental en robotique mobile.

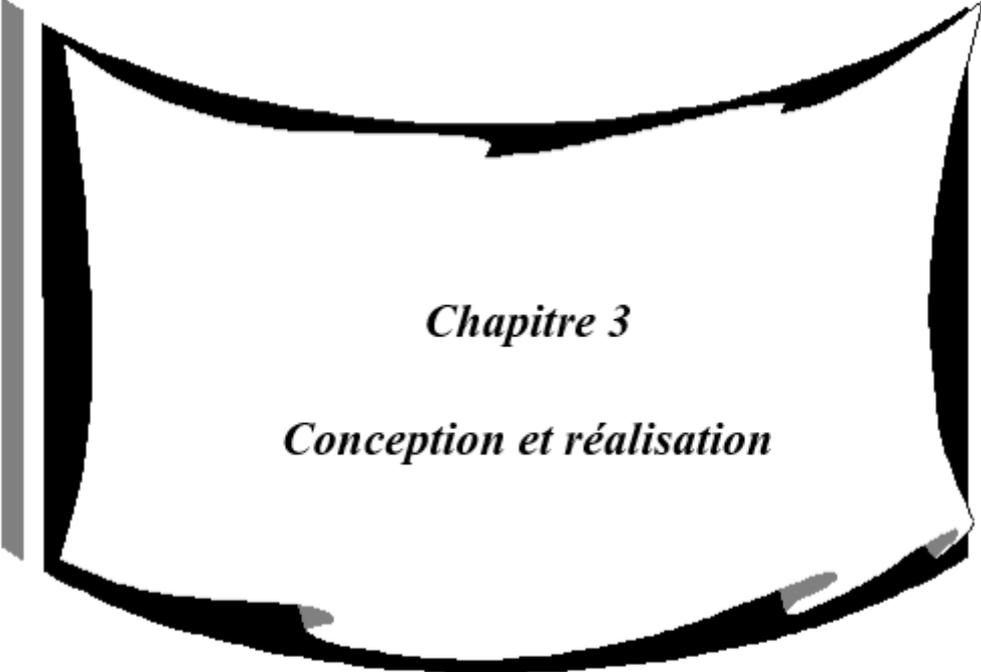
Elle permet en particulier de construire une carte utile dans de nombreuses applications (reconnaissance, sécurité, assistance, etc). La création d'une carte nécessite la résolution de trois sous-problèmes, la localisation du ou des robots, la cartographie des zones explorées et l'exploration, au sens de la découverte d'un espace inconnu. Nous nous intéressons dans cette partie à ce dernier problème. [**Frédéric, 13**]

Les problèmes d'exploration peuvent être classés selon plusieurs aspects : la connaissance de l'environnement, le type d'environnement, le synchronisme des opérations faites par les entités, le nombre d'entités en action, les qualités propres aux entités (p. ex. la capacité sensorielle d'un type d'entité), etc. Pour les besoins du sujet du présent mémoire, à savoir l'exploration par deux agents mobiles d'un environnement simple et connu, l'état des connaissances sera présenté en utilisant les trois aspects suivant : la connaissance de l'environnement, le nombre d'entités en action, ainsi que le type d'environnement. La revue de littérature qui suit divisera les travaux connexes à ce mémoire, d'abord selon le niveau de connaissance disponible sur l'environnement et ensuite selon le nombre d'entités en action pour exécuter l'exploration. Finalement, la littérature pertinente sera divisée à nouveau selon le type d'environnement (soit un terrain géométrique, soit un graphe).

III. Conclusion

Dans ce chapitre, on a basé beaucoup plus sur un état de l'art sur les algorithmes de couverture robot. Nous avons commencé par une définition de couverture, puis nous avons présenté les types des algorithmes existants qui sont classés en général en deux groupes : les algorithmes de couverture pour un seul robot et les algorithmes de couverture pour multi robots, à partir de ces deux types des algorithmes nous avons présenté les plus importants algorithmes de chaque type, en passant par un rappel sur la définition de méthode Spanning trees (ST), puis les définitions et principes des algorithmes Spanning Tree Couvrant (STC) avec ses trois versions (STC off-line Algorithme STC On-line Algorithme STC Ant-like), passant par Algorithme Spiral, ensuite nous avons présenté les principaux des algorithmes MSTC et MRSAM et MCF . Puis nous avons identifié Les champs de potentiels artificiels, Enfin nous avons terminé par une comparaison des résultats des algorithmes de couverture mono-robot et multi-robots.

L'efficacité des algorithmes d'exploration est souvent évaluée par rapport au chemin le plus court menant au but désiré. Un algorithme d'exploration est dit optimal si le coût du chemin produit par ce dernier n'est pas plus élevé que celui du chemin visitant tous les endroits à explorer à coût minimal. D'autre part, un algorithme d'exploration est dit « compétitif », si le ratio du coût de ce dernier par rapport au plus petit coût d'un tel chemin est borné par une constante



Chapitre 3

Conception et réalisation

Introduction

I. Conception

I.1. Objectifs de l'application

I.2. Modélisation du problème

I. Réalisation

I.3. Comportement des agents

II. Réalisation

II.1. Environnement de programmation

II .2. Présentation de l'application

II.3. Résultats expérimentaux

III.1. Comparaison de notre algorithme (SMRSA) avec celui de MRSAM

IV. Conclusion

Introduction

La résolution collective de problème doit beaucoup à l'analyse des techniques de couverture chez les animaux ou les humains, tâche qui consiste à rechercher des cibles dans des environnements inconnus à l'avance et dans des positions inconnues. Notre objectif est résoudre le problème de couverture mono et multi robots en utilisant une hybridation de deux algorithmes très connus le premier est utilisé pour assurer les points les plus proches à la base, c'est la division en rectangle avec des longueurs différentes [Sarid, 2011], et le deuxième c'est la couverture des rectangles par un nouveau algorithme qui a été proposé Spiral MRAD (spiral Multi Robot Area Decomposition).

Notre étude se base sur un ensemble de simulations et de mesure de performances pour la résolution de ce problème. Le système simulé est un ensemble homogène d'agents réactifs exécutant un comportement purement réactif complété d'actions de coopérations.

Ce chapitre se compose de deux parties. La conception et la modalités des différents constituants de notre système multi agent réactif et le nouveau algorithme des agents réactifs font l'objet de la première partie de ce chapitre, et les simulations en faisant varier les différents paramètres (nombre d'agent, taille de l'environnement, quantité des obstacles, longueur des rectangles et position de la cible) font l'objet de la deuxième partie du chapitre, nous terminerons ce chapitre par des discussions sur les résultats.

II. Conception

I.1. Objectifs de l'application

La question centrale dans ce mémoire c'est la résolution du problème de couverture avec un ou plusieurs robots dans un environnement inconnu, qui représente un benchmark du domaine. Plusieurs objectifs doivent être réalisés afin de répondre à cette question, parmi ces objectifs on citera :

1. Utiliser un type d'agents aussi simple que possible (agents réactifs coopératifs) ;
2. Utiliser une méthode stratégique pour la marche des agents (spirale) ;
3. Faire émerger des chemins de plus courts distance entre la source et la base, (en utilisant des champs de potentiels artificiels (APF)) ;
4. Etudier les paramètres induit à des meilleures performances (nombre d'agents, taille de l'environnement, largeur des rectangles);

5. Evaluer la robustesse du nouveau comportement au bruit dans la navigation (quantité d'obstacles);

I.2. Modélisation du problème

I.2.1. Modélisation de l'environnement

Nous considérons des environnements qui contiennent un ou plusieurs cibles dispersées dans des positions inconnues de l'environnement. L'environnement est modélisé par une grille qui se compose d'un ensemble de cellules, chaque cellule est peut-être :

- Un obstacle (couleur noir) ;
- Contient une cible (couleur verte) ;
- Est la base (couleur rouge), à partir de laquelle tous les agents commencent la couverture, elle se positionne au centre de l'environnement ;
- Contient un agent, dont chacun est identifié par un numéro unique ;
- Contient une marque (couleur gris foncé) ;

L'environnement est divisé en un ensemble de rectangles de longueur variable, l'objectif de cette division est d'assurer la couverture des endroits les plus proches à la base dans un premier temps, puis le passage à la découverte des cellules qui sont plus loin de la base, cette technique est connue sous la technique de central place foraging.

I.2.2. Modélisation des agents

Les agents sont des agents réactifs et coopératifs, ils occupent une cellule et sont capables de percevoir les quatre cellules voisines, pour des raisons de simplicité et d'optimalité, nous permettant aux agents de se trouver en même temps dans la même cellule. Chaque agent est défini par les capacités suivantes :

- Lecture et écriture des valeurs entières dans la cellule dans laquelle il se trouve ;
- Percevoir les quatre cellules voisines ;
- Déplacer à une cellule voisine si elle n'est pas un obstacle ;
- Détecter la présence des cibles ou d'obstacles dans les quatre cellules voisines ;
- Coloration et effacement des cellules par des couleurs spécifiques pour garder trace des cellules de commencement du spiral ;

I.3. Comportement des agents

Le modèle comportemental des agents est limité par certaines conditions relatives à l'environnement dans lequel ils évoluent. Ces comportements sont issus de leur stimulus local. Le comportement de nos agents fait référence au comportement de MRSAM [Sarid, 2011], ce dernier a été modifié pour assurer plus de recrutement et donc plus de performances en temps, plus de robustesse au bruit et plus d'adaptation;

Nous en avons déduit le comportement générique de chaque agent, exprimé par le graphe d'état de la figure suivante:

Chaque agent va choisir l'un ou l'autre comportement selon le percept qu'il reçoit, l'algorithme qui lui permet ce choix est donné comme suit (Algorithme 1):

Algorithme 1 : Spiral MRSAD

Perceptions : 4 cellules voisines

Entrées : point de démarrage s , un ensemble n de robots;

Module Update value :

Valeur = $\min(\text{val}, 1 + \min(4 - \text{valeurs-voisines}))$.

Ecrire (update) valeur dans la cellule courante.

ALGORITHME DE RECHERCHE SPIRAL MULTIROBOT SEARCH AREA DECOMPOSITION (MRSAD)

Répéter (pour chaque robot)

Si le robot démarre à partir de la base alors

 Début

 déplacer en ligne droite et update value ;

 choisir aléatoirement une cellule libre :

 colorer cette cellule

 Fin

Sinon

 Répéter

 choisir la première cellule non colorée ;

 déplacer vers le bas, gauche, haut, droite et update value ;

 jusqu'à ce que le robot (tombe deux fois sur la frontière ou tombe sur la base ou tombe une fois

sur la base et une fois sur la frontière) ;

Revenir vers le dernier point de départ de dernier spiral en suivant les valeurs entières de l'APF déjà inscrites ;

Déplacer en ligne droite dans le sens horaire et choisir la première cellule non coloré et démarrer le spiral;

jusqu'à ce que la cible est trouvé;

on a ajouté une contrainte dans notre algorithme **Spiral MRSAD** qui s'exprime par le test suivant :

- Tester les quatre sens de zone de couverture avec la procédure ligne droite dans le cas d'un agent ou plusieurs agents [pour sortir de la première région vers la deuxième]
- si le nombre d'agent ≥ 2 on teste sur les deux sens
- si le nombre agents < 2 on teste sur les 4 sens

III. Réalisation

Dans cette partie, nous présenterons les différentes étapes de réalisation de notre projet qui a pour objectif d'implémenter un algorithme qui permet la couverture d'un environnement pour rechercher un ensemble de cibles dont les positions sont inconnus à l'avance, en utilisant seulement des agents réactifs simple, qui ont la possibilité de marqué et de coloré leurs environnement pour assurer une couverture plus rapide de l'environnement.

Nous présenterons d'abord l'environnement de programmation (Java), ensuite nous exposons les différents étapes de simulation, nous présenterons par la suite les résultats obtenus et nous terminerons par une discussion de ces différents résultats.

Avant d'aller plus loin dans les simulations, plusieurs paramètres doivent être discutés :

Choix des paramètres

Pour assurer l'efficacité et la robustesse de notre proposition, plusieurs paramètres doivent être traités, ces paramètres influence de façon directe les résultats obtenus, il y a pas de méthode claire qui permet la fixation de ces paramètres, et donc une suite de simulation basé sur la variation à la fois de chacun de ces paramètres sera discuté plus loin dans ce chapitre

pour déduire quel est le paramètre qui influencera le résultat et quelle est la valeur adéquate pour que le résultat soit optimale la suite de ces paramètres est donné comme suit :

- **La taille de l'environnement** : Nous avons utilisés des environnements carrés avec une taille variable ;
- **La densité des obstacles (pourcentage)**:est aléatoire, et sont disséminés de façon aléatoire dans l'environnement, un choix au préalable du pourcentage des obstacles doit être fixé avant la simulation. Durant les simulations, on va consacrer toute une partie sur la variation de ce paramètre et son influence sur la robustesse de notre algorithme ;
- **Le nombre de cibles** : c'est fixé selon le choix de l'utilisateur.
- **Le nombre des agents** : le choix ce fait de façon aléatoire, les agents sont homogènes, et ils commencent tous de la base, l'agent a la possibilité de porter à la fois une quantité $Q_{\text{limite-source}}$, c'est un paramètre très importante dans nos simulations.

III.1. Environnement de programmation

Parler de l'implémentation revient à détailler l'aspect matériel, l'environnement de développement et les différents outils qui ont été utilisé pour réaliser l'application.

1) Aspect matériel

Notre projet a été développé sur un pc :

- + Type : Presario CQ56 Netbook PC
- + Processeur : Pentium Dual-Core CPU T4500 @ 2.30GHz 2.30GHz.
- + RAM : 2,00 GO
- + Disque dure : 395,9 GO

2) Environnement de développement

L'application réalisée a été implémentée avec le langage **JAVA version 4.2** sous l'environnement **Eclipse JUNO (version 2013)**.

Pour quoi Java ?

Contrairement à d'autre langage, la simplicité de java est en fait directement liée à la simplicité du problème à résoudre. Ainsi, un problème de définition de l'interface utilisateur peut être un véritable casse-tête avec certains langages, avec java, créer des fenêtres, des boutons et des menus est toujours d'une simplicité extrême, ce qui laisse au programmeur la possibilité de se concentrer sur son problème à résoudre.

- Java est un environnement complet qui offre des flexibilités et des possibilités importantes ;
- Java est un langage relativement simple à apprendre et lire ;
- Java offre une portabilité grâce à la machine virtuelle (JVM) ;
- très adéquat pour le développement des agents comme des objets autonome et mobile.
- C'est un langage orienté objet dérivé du C, mais plus simple que C.
- Il est multiplateforme : tous vos programmes tournent sans modification sur toutes les plateformes où existe Java.
- Il est doté en standard d'une riche bibliothèque de classes, comprenant la gestion des interfaces graphiques (fenêtres, boîtes de dialogue, contrôles, menus, graphisme...), la programmation multithreads (multitâches), la gestion des exceptions, les accès aux fichiers et au réseau (notamment Internet)... etc.

III.2. Présentation de l'application

L'interface graphique principale de cette application englobe un ensemble de composants à partir des quels on peut exploiter notre système, la figure ci-dessous montre cette interface et ses différentes parties.

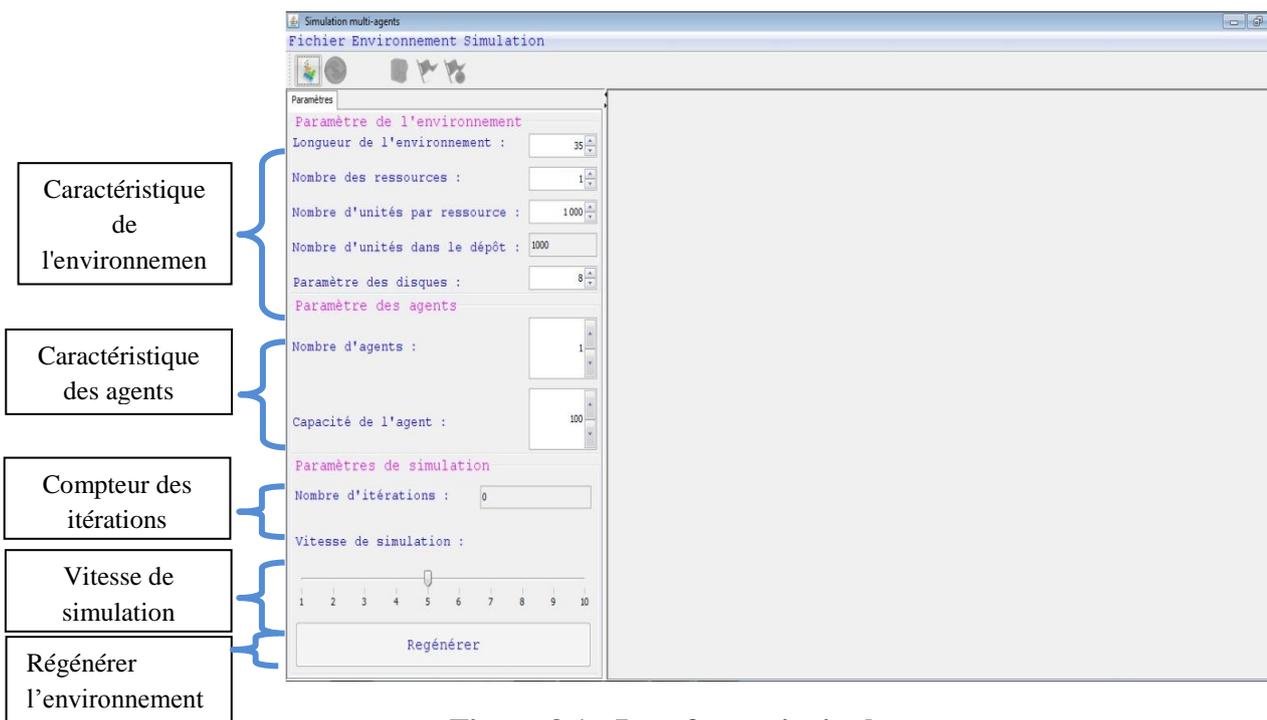


Figure 3.1 : Interface principale

L'exploitation de notre application est très simple, la fenêtre se compose de deux menus : le menu fichier permettant la création de l'environnement et de quitter l'application (Figure IV.3); et le menu simulation permettant de lancer la simulation (avec les paramètres déjà choisis), de suspendre, de reprendre ou de lancer une nouvelle simulation

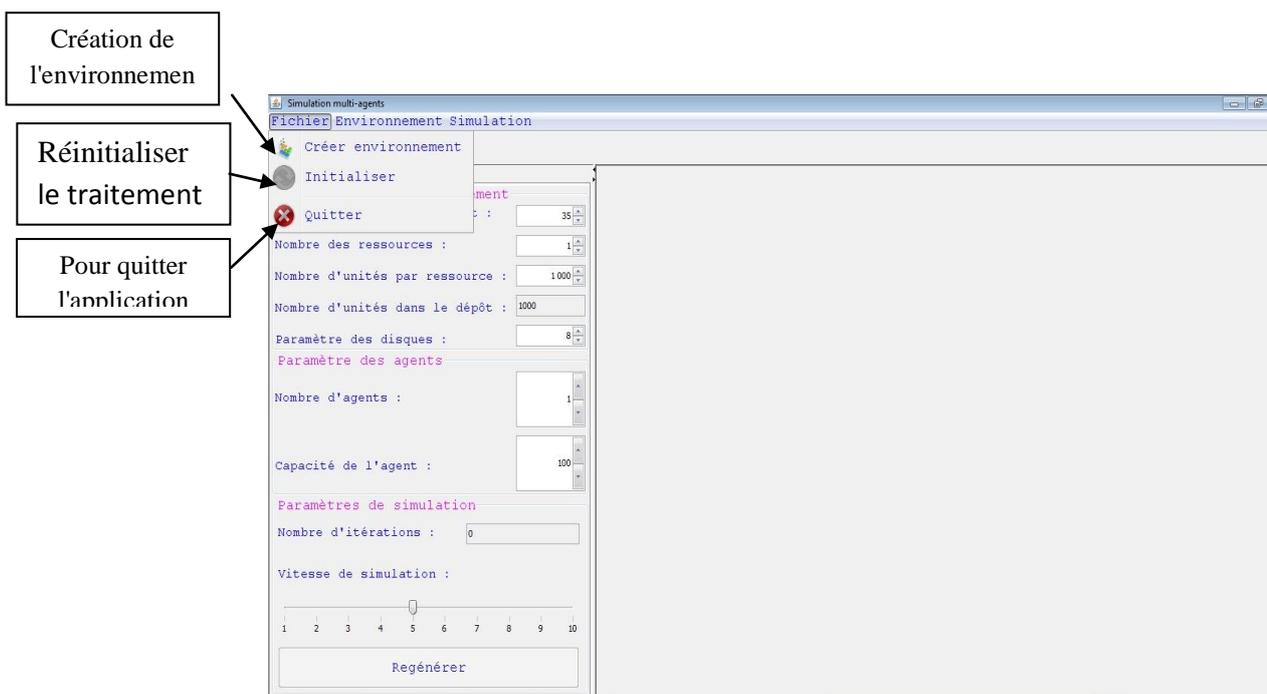


Figure 3.2 : Création d'environnement

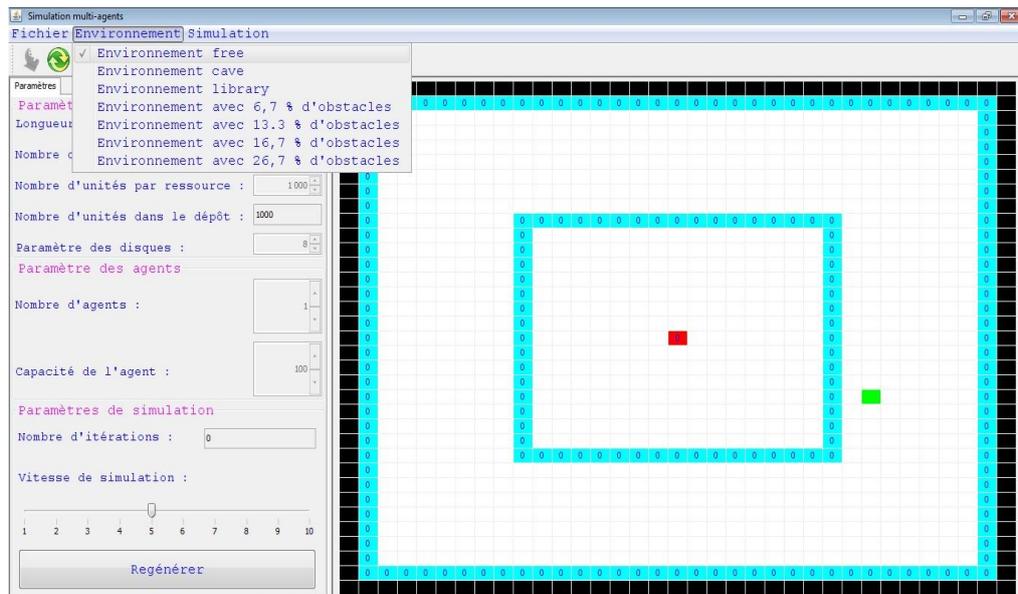


Figure 3.2 : Création d'environnement(a)



Figure3.2 : Création d'environnement(b)

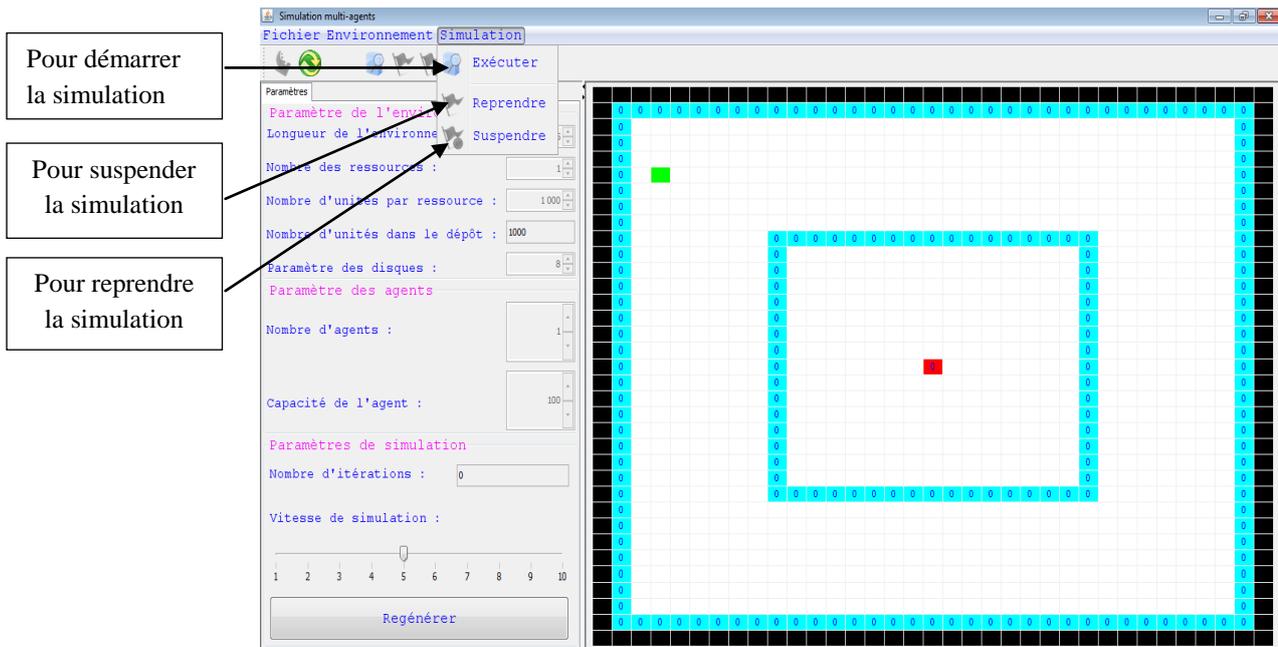


Figure 3.3: Lancer la simulation

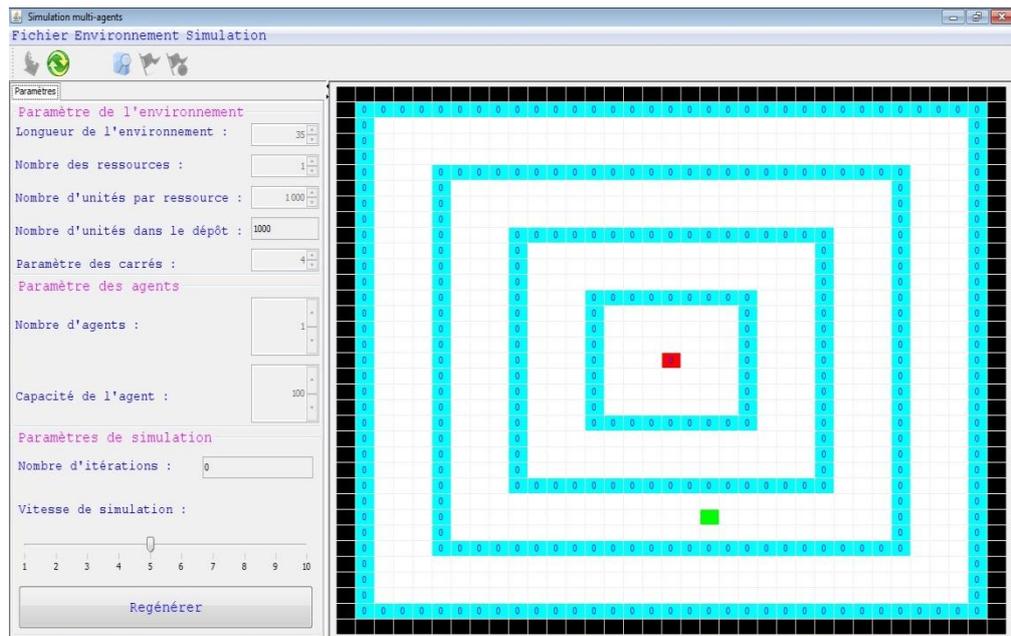


Figure 3.4: Division de l'environnement selon le paramètre de rayon de carré

La création de l'environnement c'est-à-dire la fixation des différents paramètres dans la figure 3.5 Doit être faite avant de lancer la simulation.

III.3. Résultats expérimentaux

Plusieurs configurations ont été testés pour déduire le quels des paramètres cités auparavant influencera la robustesse et l'efficacité de notre algorithmme. Après les différents tests nous

avons conclu que les paramètres qui ont vraiment une influence significative sur l'efficacité de notre algorithme sont la taille de l'environnement, le nombre d'agents et pour l'environnement free, donc une ébauche sur ces deux paramètres est donnée dans ce qui suit :

Configuration 1:

- Environnement de 40 x 40 cellules, 1 cellule est une cible, qui est également réparti de façon aléatoire.
- Le nombre d'agents est fixé à 10.

Les figures qui se suivent montrent l'environnement de la configuration 1 et les différentes étapes de la simulation :

a) Lancer une simulation avec un agent et une source :

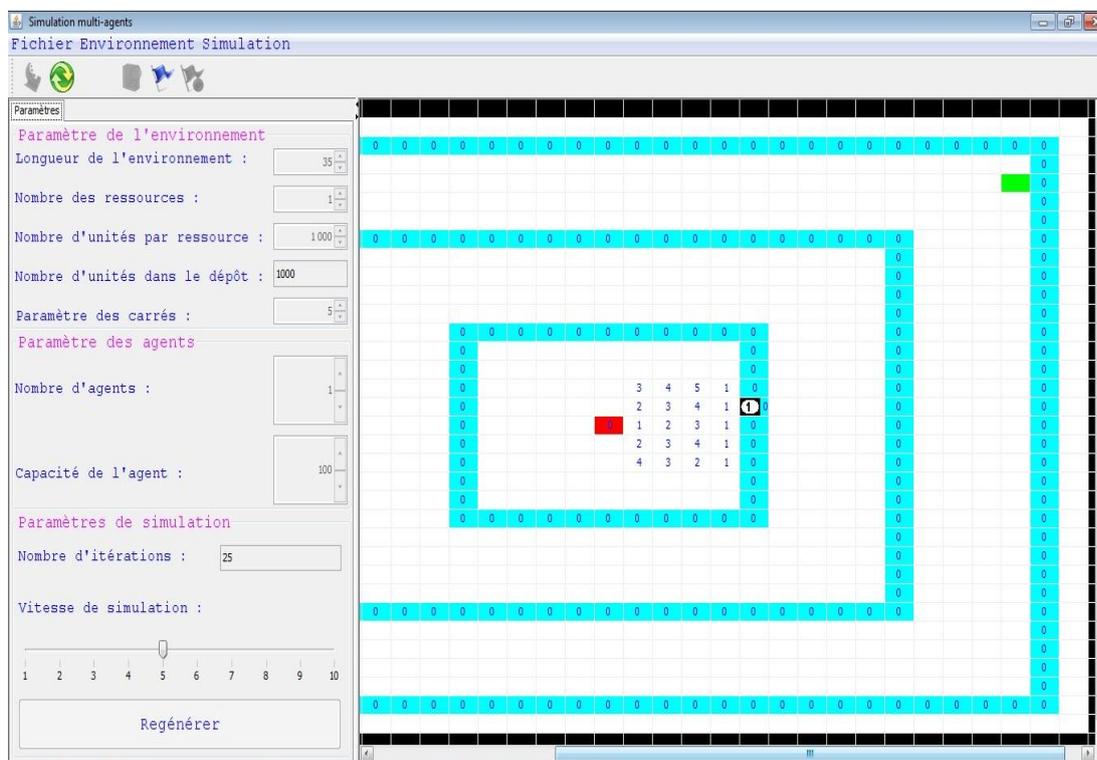


Figure 3.5 : (Etape 1) lancer la couverture mono-robot

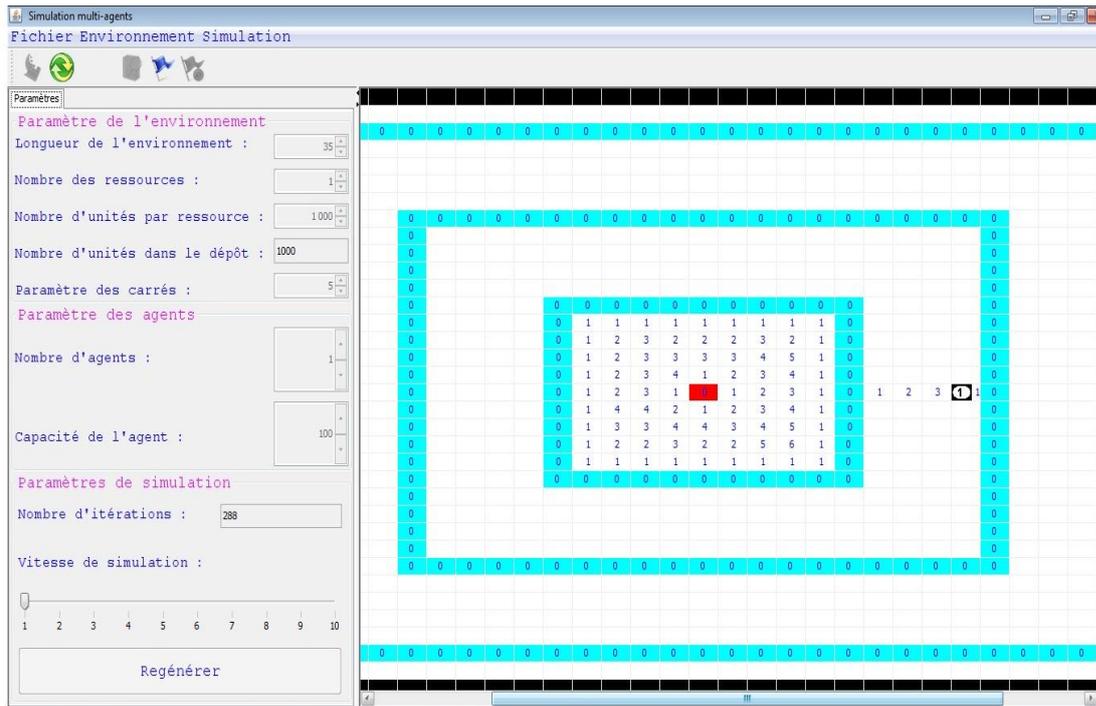


Figure 3.5: (Etape 2) déplacement de robot vers la deuxième région

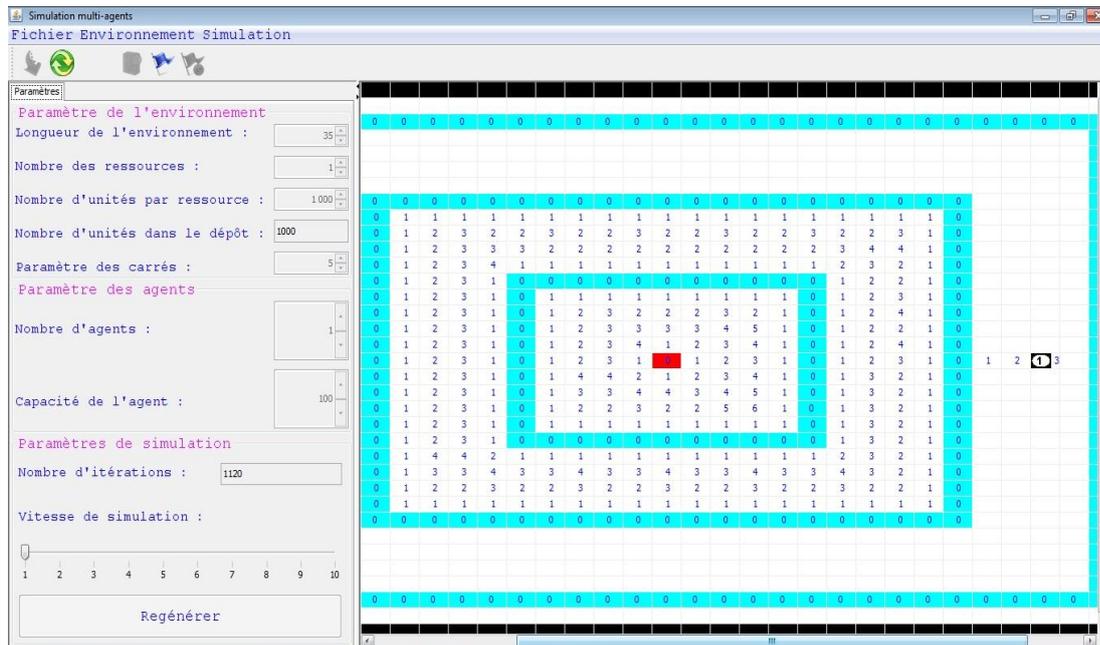


Figure 3.5: (Etape 3) déplacement de robot vers la troisième région

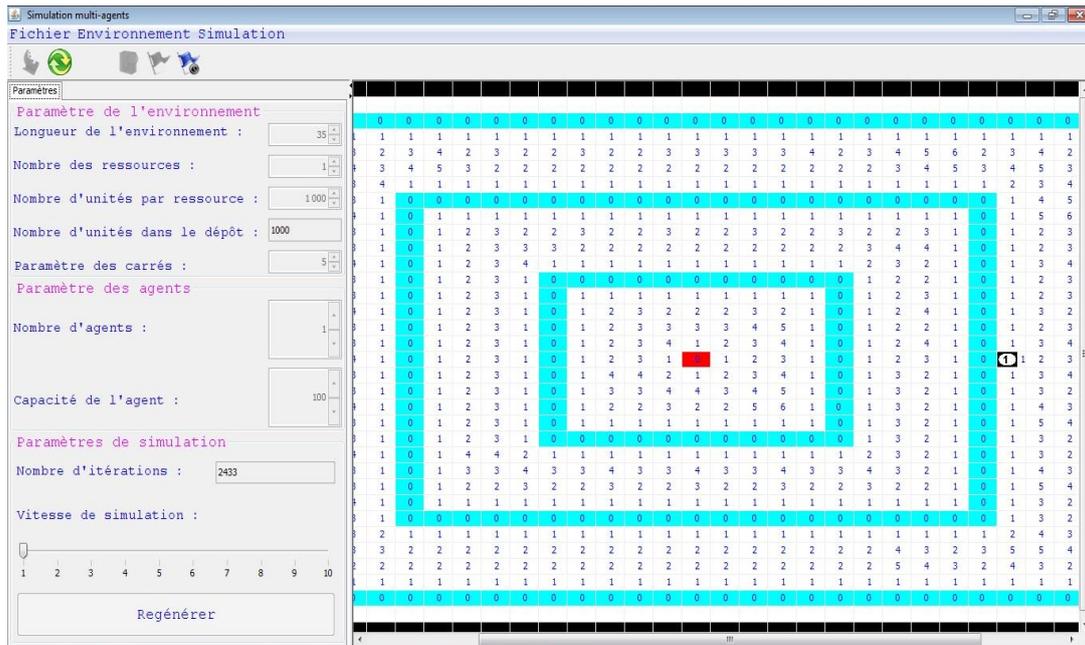


Figure 3.5: (Etape 4) couvrir la quatrième région

b) Lancer une simulation avec quatre agents et une source :

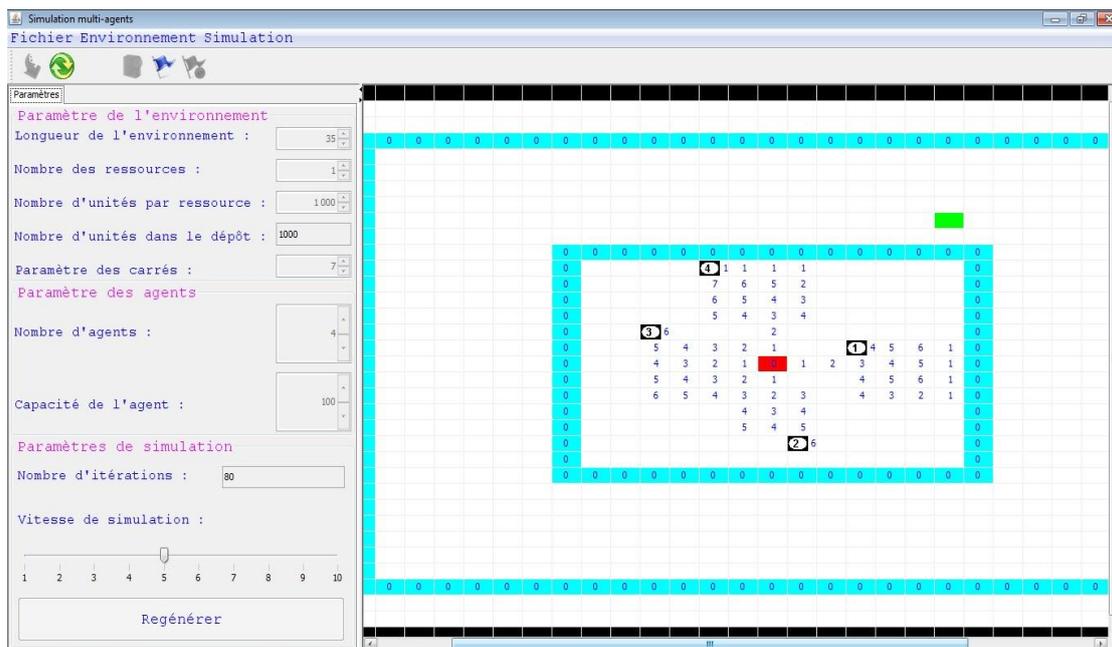


Figure 3.6 : (Etape 1) lancer la couverture multi-robot

c) Lancer une simulation avec 20 agents et une source :

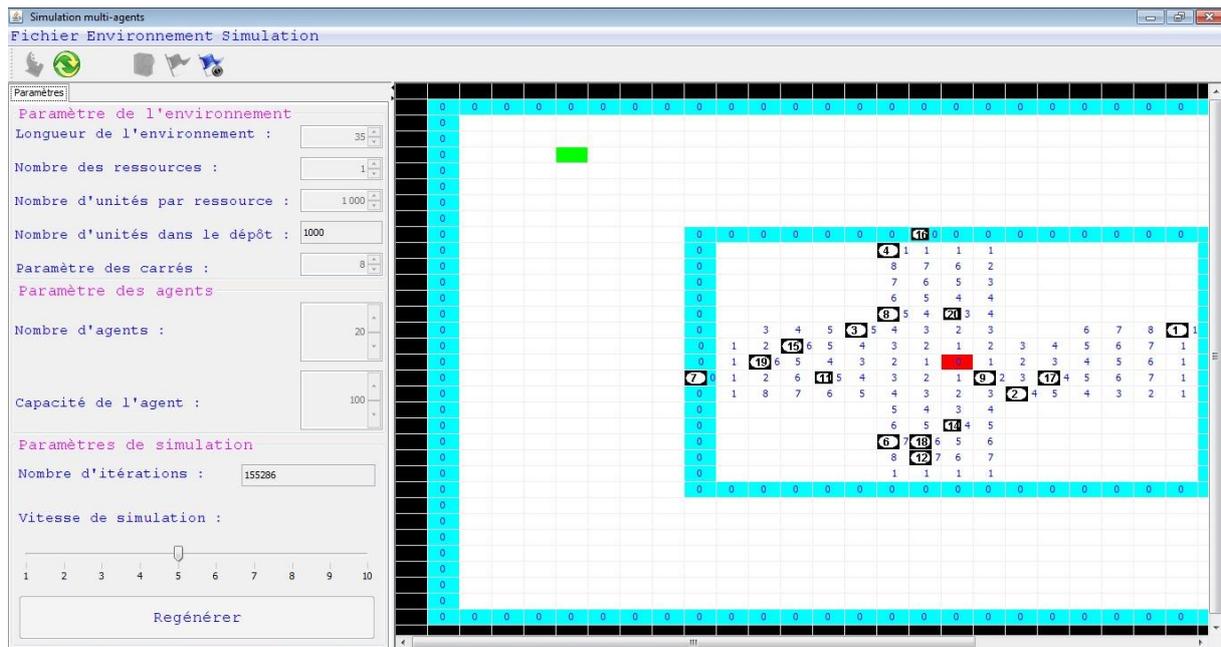


Figure 3.6 : (Etape 2) lancer la couverture multi-robot

Discussion des résultats

Une simulation consiste pour chaque agent à lire ou écrire une valeur sur sa cellule actuelle et de passer à une cellule voisine. Nous avons répété chaque simulation plusieurs fois dans chaque configuration (jusqu'à 20), les environnements sont définis comme des carrés. La base est toujours située dans le centre (avec une couleur rouge) et tous les agents (avec des couleurs bleus) part d'elle. Obstacles (avec une couleur noire) et lieux ressources (avec une couleur verte) sont réparties au-hasard dans l'environnement.

Influence de la taille de l'environnement sur les performances :

On voit que la taille de l'environnement influence les performances de l'algorithme lorsque le nombre des agents est variable. Évidemment, un environnement large, nécessite plus de temps de couverture et d'exploration.

On observe que le nombre d'itération augmente en parallèle avec l'augmentation de la taille de l'environnement ainsi avec l'augmentation de nombre des régions en carré. En d'autres termes, le temps d'exploration devient exponentiel. Est la marche de l'agent pseudo-spiral reste efficace quel que soit la taille de l'environnement.

Influence de nombre des agents sur les performances :

On voit que le nombre des agents influence les performances de l'algorithme lorsque :

1. Le nombre des agents < 4 → Un environnement large nécessite plus de temps de couverture et d'exploration.
2. Le nombre des agents $= 4$ → un environnement large nécessite moins de temps de couverture et d'exploration.
3. Le nombre des agents > 4 → un environnement large nécessite minimum de temps de couverture et d'exploration.

IV. Conclusion

Dans ce chapitre, nous avons présenté en détail notre application à savoir le modèle comportementale de nos agents réactifs et coopératifs, ainsi que l'algorithme leurs permettant la sélection des comportements adéquates selon le stimulus reçus, les agents sont capable de construire des chemins optimaux entre la source et la base (construction de l'APF), cela leurs permet de revenir rapidement et sans se coincer dans des minimas locaux de l'environnement, d'un autre coté les agents peuvent coloré les chemins entre la base et la source pour permettre un retour aux sources déjà découverte.

La suite des simulations faite montre que notre algorithme (SMRSA) proposé après l'amélioration de l'algorithme original (MRSAM) fournit des résultats efficaces que celle de l'algorithme original(MRSAM).

D'autres problèmes ont été découverts lors de l'implémentation et qui font l'objet de nos perspectives.

Conclusion Générale et perspectives

Les travaux de coordination multi-agent se répartissent en deux catégories. La première catégorie regroupe les aspects de la coordination vue sous l'angle de la coopération. Ces travaux font l'hypothèse que les agents sont motivés collectivement à l'accomplissement d'un but commun. C'est le cas par exemple en planification distribuée ou dans les algorithmes de recherche distribuée. La seconde catégorie regroupe les travaux considérant que les agents sont motivés individuellement à l'accomplissement de buts qui leur sont propres tout en essayant de conserver certaines propriétés au niveau du groupe. Notre travail s'intègre dans les deux catégories.

Nous avons abordé dans ce papier le problème de l'exploration multi-robot d'un environnement inconnu. Nous avons proposé un nouvel algorithme (Spiral MRSAD) pour la couverture d'un environnement que découvrent les robots. Celui-ci repose sur la notion de spirale vers un but. Les robots sont alors répartis sur les régions selon leurs meilleures positions pour commencer la couverture. L'évaluation des positions est fondée sur le marquage de chaque cellule visitée.

Dans ce mémoire nous avons concentré sur un algorithme qui a été déjà proposé dans le cadre de la couverture multi-robots, celui de MRSAM. Ce dernier, permet la décomposition de l'environnement en un ensemble de disques et chaque robot va exploiter un disque différent.

Lors de l'implémentation dans un environnement libre un problème a été découvert, et qui fait l'objet de notre perspective future :

Problème trouvé :

- Parfois l'agent n'arrive pas à couvrir toutes les cases de première région et passe à la deuxième région. Pour cela on a proposé une solution efficace ;

Solution proposée :

- on a ajouté une contrainte dans notre algorithme **Spiral MRSAD** qui s'exprime par le test suivant :
 - Tester les quatre sens de zone de couverture avec la procédure ligne droite dans le cas d'un agent ou plusieurs agents [pour sortir de la première région vers la deuxième]
 - si le nombre d'agent ≥ 2 on teste sur les deux sens
 - si le nombre agents < 2 on teste sur les 4 sens

Bibliothèques :

[Adina, 02] : Adina Magda Florea « Agents et Systèmes Multi-agents » cours, l'Université Politehnica de Bucarest ,2002

[Arnaud ,11] : Arnaud Glad «Etude de l'auto-organisation dans les algorithmes de patrouille multi-agent fondés sur les phéromones digitales » thèse pour l'obtention du Doctorat de l'université Nancy 2, 2011.

[Bautin et al, 11] : A. Bautina, O. Simonin et F. Charpilleta , « Stratégie d'exploration multi-robot fondée sur les champs de potentiels artificiels», Manuscrit auteur, publié dans "Journées Francophones sur les Systèmes Multi-Agents (JFSMA) ,2011.

[Bond et al, 88]. Bond A. et Gasser L., « Readings in Artificial Intelligence». Morgan Kaufmann, 1988.

[Brian, 97] : Brian Yamauchi. A frontier-based approach for autonomous exploration: Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, page 146, Washington, DC, USA, 1997.

[Brooks 1986]: R. Brooks, « a Robust Layered Control System for a Mobile Robot » IEEE Journal of Robotics and Automation

[Champion, 03] :A. Champion. « Mécanisme de coordination multi –agent fondé sur des jeux: application à la simulation comportementale de trafic routier en situation de carrefour». Thèse de doctorat, l'université de valenciennes et du hainaut-cambrésis, 2003.

[Courdier , 07] : Rémy Courdier , «Systèmes Multi-Agents Partie 2 Agents et Systèmes Multi-Agents», Université de la Réunion , 2007

[Decker, 95] : Decker K., «*Environment centred analysis and design of coordination mechanisms*», University of Massachussets, 1995

[Deloach 09]: Deloach.S.A. « Organizational model for adaptive complex systems ». In Virginia Dignum (ed.) Multi-Agent Systems: Semantics and Dynamics of Organizational, University, USA (2009).

[Doran et al. 97] : Doran.J. E., Franklin.S., Jennings.N.R., Norman.T.J. «On cooperation in multi-agent systems ». The Knowledge Engineering Review .1997

[Drogoul, 93] : A. Drogoul. « De la simulation multi agent à la résolution de problèmes. Une étude de l'émergence de structures d'organisation dans les systèmes multi agents ». 1993.

[Durfee et al , 89] : Durfee E.H., Lesser V. et Corkill D., Trends in cooperative distributed problem solving. *IEEE Knowledge and Data Engeneering*, 1989.

[Ferber, 95] : J. Ferber. « Les systèmes multi-agents vers une intelligence collective », inter éditions, 1995.

[Ferber et al, 09]: Ferber.J., Stratulat.T., Tranier.J, « Towards an integral approach of organizations in multi-agent systems: the MASQ approach», university de Montpellier, France, 2009

[Frédéric, 13] : Frédéric Lessard « exploration optimale d'un segment de droite par deux agents mobiles», mémoire présenté comme exigence partielle de la maîtrise en informatique université du québec en outaouais, 2013.

[Gabriely and Al, 99]: Yoav Gabriely and Elon Rimon « Spanning-Tree Based Coverage of Continuous Areas by a Mobile Robot» Department de Mechanical Engineering Technion, Israel Institute of Technology, 1999

[Glizes, 04]: Gleizes.M.P, «Vers la résolution de problèmes par émergence». Thèse d'habilitation de l'Université de Toulouse 2004.

[Glizes et al. 99]: Gleizes.M.P., Camps.V., Glize.P. «A theory of emergent computation based on cooperative self-organization for adaptive artificial systems». Institut de Recherche en Informatique de Toulouse, France, 1999.

[Haddad et Hamidi, 09] : A. Haddad et M. Hamidi. « Modélisation du contexte avec prise en compte d'incertitudes pour les applications sensibles au contexte (cas : les environnements intelligents) ». Mémoire de fin d'études Pour l'obtention du diplôme d'ingénieur d'Etat en génie : informatique, Alger, 2009.

[Jacky et al, 04] : H. Jacky Chang, C. S. George Lee, Yung-Hsiang Lu, and Y. Charlie Hu «Energy-Time-Efficient Adaptive Dispatching Algorithms for Ant-Like Robot Systems »School of Electrical and Computer Engineering Purdue University, 2004.

[Jiquan et al, 08]: Jiquan.S., Zhiqiang.W. «Service-Oriented Organization Management and Coordination Control of MAS». In IEEE International Conference on Intelligent Computation Technology and Automation, (2008).

[Khatib, 1986]: Oussama Khatib, «*Real-time obstacle avoidance for manipulators and mobile robots*» Article; International Journal of Robotics Research, l'Université Stanford, 1986.

[Kaplan, 1998] : F. Kaplan «Role de la simulation multi-agent pour comprendre l'origine et l'évolution du langage», nom de journal: BARTHÈS, J.-P., CHEVRIER, V., et BRASSAC, C., éditeurs, Systèmes multi-agents: de l'interaction à la socialité (JFIADSM98), 1998.

[Labidi et al, 93] : S. Labidi, W. Lejouad «De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents »Institut national de recherche en informatique et en automatique, 1993

[Lopez et al, 97] R. Lopez De Mantaras, J. Amat, F. Esteva, M. Lopez, and C. Sierra. Generation of unknown environment maps by cooperative low-cost robots. 1997

[M.Gouasmi, 05] : M.GOUASMI Ingénieur d'état en informatique Option systèmes d'informations avancés «Intégration d'ontologie dans les actes de communication inter-agents» Université Ibn Khaldoun de Tiaret, 2005

[Melaye et al, 05] : Melaye.D., Demazeau.Y. «Modèles et Réseaux de confiance Droits et Devoirs d'Agents Autonomes Confiance et modèles de confiances pour la régulation des droits et devoirs d'agents autonomes» Laboratoire Leibniz, Equipe MAGMA, 2005.

[Moujahed, 07] S. Moujahed. « Approche multi-agent auto-organisée pour la résolution de contraintes spatiales dans les problèmes de positionnement mono et multi-niveaux ». Thèse de l'université de Technologie de Belfort-Montbéliard et de l'Université de Franche-Comté, 2007

[Noam et al, 05]: Noam Hazon and Gal A. Kaminka «Redundancy, Efficiency and Robustness in Multi-Robot Coverage »

[Nembrini et al, 07] : Julien Nembrini & Alcherio Martinoli «Robotique en Essaim,Récents Résultats et Directions Futures»Ecole Polytechnique Fédérale de Lausanne, 2007

[Sahki, 08] : Mr Hanine Sahki, « Vers une Architecture d'un Système de

Dialogue Multi Agents Basé sur l'Argumentation Application à la négociation dans le domaine de e-commerce», Mémoire présenté en vue de l'obtention du Magister en Informatique, 2008

[Sarid et Shapir, 06] : Shahar Sarid and Amir Shapiro «MRSAM: A Quadratically Competitive Multi Robots Navigation Algorithm»Department of Mechanical Engineering Ben Gurion University of the Negev, Israel, 2006.

[sarid, 11] : Shahar Sarid, « Heterogeneous Multi-Robot Search Algorithms», University de Negev, 2011.

[Shehory et al, 98] : Shehory O., Sycara K., Chalasani P. et Somesh J., Increasing resource utilisation and task performance by agent cloning, *Proceedings of the ICMAS, ATAL 98*,

[Panait et al, 05] : Panait.L., Luke.S. «Cooperative Multi-Agent Learning: The State of the Art. In Autonomous Agents and Multi-Agent Systems», University de George Mason. 2005.

[Russel, 06] : Russell.S, «Intelligence artificielle». Pearson Education 2e édition. 2006

[Sabouret, 09] : Sabouret.N. «Interactions sur le fonctionnement dans les systèmes multi-agents ouverts et hétérogènes». Habilitation à Diriger des Recherches, Université Pierre & Marie Curie 2009.

[Thomas, 05] : Thomas.V, «Proposition d'un formalisme pour la construction automatique d'interactions dans les systèmes multi-agents réactifs». Thèse de doctorat de l'université Nancy 1, 2005.

[Tom et al, 97] : Tom Duckett and Ulrich Nehmzow. Experiments in evidence based localisation for a mobile robot. In Proceedings of the AISB Workshop on Spatial Reasoning in Mobile Robots and Animals, Manchester, UK, 1997.

[Velagapudi et al, 07] : Velagapudi.P, Prokopyev.O, Sycara.K., Scerri.P. «Maintaining Shared Belief in a Large Multiagent Team». In Proceedings of FUSION, 2007.

[Wesley, 01]: Wesley H. Huang, «Optimal Line-sweep-based Decompositions for Coverage Algorithms» Department of Computer Science Rensselaer Polytechnic Institute Troy, New York, 2001.

[Xiaoming et al, 05] : Xiaoming Zheng ,Sonal Jain ,Sven Koenig ,David Kempe «Multi-Robot Forest Coverage» Department of Computer Science University of Southern California Los Angeles, 2005

[Yoav et al, 02]: Yoav Gabriely and Elon Rimon «Spiral-STC: An On-Line Coverage Algorithm of Grid Environments by a Mobile Robot» Technion, Israel Institute of Technology, 2002.

[Yoav et al, 99]: Yoav Gabriely and Elon Rimon, «Spanning-Tree Based Coverage of Continuous Areas by a Mobile Robot», Dept. of Mechanical Engineering Technion, Israel Institute of Technology, 1999

