الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université de 8 Mai 1945 – Guelma -
Faculté des Mathématiques, d'Informatique et des Sciences de la matière
Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Systèmes Informatiques

Thème :

Vers une stratégie de recherche de robotique mobile adaptative

Encadré Par :

Dr. ZEDADRA Ouarda

Présenté par :

CHIHEB Badreddine

Juillet 2019

Dédicace

A mon PERE

A ma MERE

A mes FRERES et SŒURS

A ma femme et sa famille

A toute la famille

A mes professeurs

A toute la promotion 2019

A tout mes amis et collègues

A tous ce que j'aime et tous ceux qui m'aiment

CHIHEB Badreddine

Remerciement

Au terme de ce mémoire que nous achevons grace à Dieu tout puissant.

Mes sincères remerciement et ma reconnaissance vont à tous ceux qui m'ont aidé tout au long de mon parcours

A Mme ZEDADRA Ouarda pour son assistance, sesprécieux conseils et sa disponibilité constante ainsi que le savoir qu'elle m'a transmis.

Je remercie également l'honorable jury pour l'intérêt qu'il porte à juger notre travail.

CHIHEB Badreddine

Résumé

Le déploiement des travaux existantes du foraging dans le monde réel reste encore un défi, du fait que ces travaux ne considèrent pas assez de caractéristiques importantes qui répondent aux exigences du monde réel tel que : l''adaptation en ligne, la durabilité du système et l'adaptabilité dans des scénarios critiques.

Dans le but de rapprocher le maximum possible des applications du monde réel, une nouvelle instance du foraging a été créé. Cette instance prend en considération la caractéristique de durabilité du système. Dans cette nouvelle version les robots sont recensés satisfaire les demandes qui proviennent de l'extérieure et donc de maintenir la durabilité du système ou encore la survie des dépôts.

Nous avons proposé un algorithme de foraging multi-dépôts, dans lequel les robots répondent aux exigences de systèmes externes, permettant ainsi de maintenir la durabilité ou la survie du système. L'algorithme proposé utilise : (1) Une stratégie d'exploration basé abeilles pour explorer et rechercher les objets ; (2) un modèle simple d'interaction pour recruter des robots étrangers en cas d'alerte.

L'algorithme proposé a été testé dans plusieurs configurations environnementales et il a donné de bons résultats, en sens que les robots ont assurer la survie du système, soit avec la stratégie d'exploration seulement ou avec le processus de recrutement en cas d'alerte.

Mots-clés: Intelligence en essaim, Robotique en essaim, Foraging Multi-Robots, Durabilité, Lévy-walk, Algorithme d'abeille.

SOMMAIRE

Chapitre 03 : Implémentation et Expérimentation

LISTE DES FIGURES

Chapitre 01 :

Chapitre 02 :

Chapitre 03 :

LISTE DES TABLEAUX

Chapitre 01 :

Chapitre 03 :

LISTE DES EQUATIONS

INTRODUCTION GÉNERALE

Introduction générale

1. Positionnement scientifique

L'intelligence Artificielle à ses débuts a puisé son inspiration dans le comportement individuel de l'être humain, en cherchant à reproduire son raisonnement. Elle s'est donc focalisée sur la manière de représenter les connaissances d'un expert et de modéliser son processus de décision, pour construire dessystèmes dont le résultat pouvait être qualifié d'*intelligent*.

L'intelligence distribuée, connu aussi sous le nom intelligence en essaim, désigne l'apparition de phénomènes cohérents à l'échelle d'une population dont les individus agissent selon des règles simples. L'interaction entre actions individuelles simples peut de façons variées permettre l'émergence de formes, organisations, ou comportements collectifs, complexes ou cohérents, tandis que les individus eux se comportent à leur échelle indépendamment de toute règle globale.

La robotique en essaim c'est une branche de la robotique mobile qui prend ses inspirations des systèmes qualifiés d'intelligence en essaim tel que les troupeaux de oiseaux, les bains de poisons, les insectes sociaux (fourmis, abeille…). Elle encourage l'utilisation de plusieurs robots avec des capacités de calcul, de perception et de mémorisation limitées. L'interaction locale entre ces individus simple produit des solution efficaces, des comportements complexes avec des couts minimales. Les problèmes résolus par des essaims de robots sont : la formation de motifs, l'agrégation, la formation de chaînes, l'auto-assemblage, le mouvement coordonné, l'évitement de trous, la recherche des objets (Foraging) et l'auto-déploiement.

Le foraging Multi-Robots constitue une métaphore pour plusieurs applications du monde réel à savoir le nettoyage, le déminage, l'exploration des environnements hostiles, la recherche et le sauvetage. Il se divise en deux catégories : foraging centralisé avec un seul dépôt (Central Place Foraging- CPF), et foraging multi-dépots (Multi-PlaceForaging- MPF) avec plusieurs dépôts. Ce dernier, est une instance qui permet de : (1) réduire les collisions au niveau du dépôt central et (2) réduire la longueur du chemin entre les ressources et le dépôt.Dans ce travail ne nous intéressons au problème de foraging multi-dépôts.

2. Problématique et objectifs

Les travaux de foraging classique ne considèrent pas assez de caractéristiques importantes qui répondent aux exigences du monde réel tel que : l''adaptation en ligne, la durabilité du système et l'adaptabilité dans des scénarios critiques. Alors que ces caractéristiques sont cruciales pour que les systèmes de foraging soient déployable sur le monde réel. Des questions qui se pose et qui permettent ainsi de considérer certaines de ces caractéristiques:

1. Est-il possible maintenir et d'augmenter la survie des dépôts en répondant à des demandes qui proviennent de l'extérieure à travers la stratégie d'exploration ?
2. Est-ce qu'une collaboration simple entre robots appartenant à des dépôts étrangers peut garantir la survie d'un dépôt en état d'alerte ?

Dans l'objectif de répondre à ces questions nous avons :

1. Proposé un système de foraging multi-dépôt qui permet de satisfaire les demandes qui proviennent à ses différents dépôts dans des périodes de temps différents. Les robots dans ce système utilisent un algorithme bio-inspiré basé algorithme de l'abeille pour l'exploration et la recherche des objets;
2. Si une alerte a été détectée dans l'un des dépôts par l'un de ses robots, ce dernier doit recruter assez de robots appartenant à des dépôts étrangers pour collaborer à maintenir la survivabilité du dépôt infecté et donc de maintenir la survie globale du système.

3. Plan de mémoire

Ce manuscrit est réparti en trois chapitres :

Le premier chapitre:nous commençons le chapitre par présenter brièvement les domaines de l'intelligence en essaim et de robotique en essaim. Par la suite, Nous définissons le problème de foraging et nous proposons une taxonomie pour la classification des systèmes de foraging existantes. Ensuite, nous synthétisons les travaux de foraging existantes, et ne les comparons qualitativement dans un tableau à travers certains critères de comparaison pertinents.

Le deuxième chapitre: ce chapitre est consacré à la conception. Nous présentons donc la problématique de recherche, les objectifs et notre proposition, Nous détaillons par la suite les algorithmes reliés. Ensuite, nous présentons l'algorithme proposé en termes d'organigramme général représentant l'algorithme, des descriptions textuelles pour expliquer les différents états et les pseudos algorithmes associés.

Le troisième chapitre: ce chapitre présente l'implémentation et les résultats obtenus par l'algorithme proposé. On commence par présenter la plateforme de simulation de robotique ARGoS, ainsi que les caractéristiques des robots utilisés (Foot-bot) est aussi faite. Par la suite, nous proposons quelques scénarios de simulation, les résultats obtenus, et les discussions.

Le manuscrit est clôturé par une conclusion récapitulative et quelques perspectives intéressantes à explorer en continuité de travail.

CHAPITRE 1

CHAPITRE I

Foraging Multi-Robots: Définitions et travaux reliés

1. Introduction

Dans la robotique en essaim, plusieurs robots individuels autonomes travaillent ensemble pour atteindre un objectif commun. L'utilisation de matériel peu coûteux et redondant dans la robotique en essaim présente des avantages évidents par rapport à l'approche traditionnelle à un seul robot pour de nombreux problèmes. Un essaim de robots peut répartir le travail entre de nombreux individus, souvent dans l'espace physique, ce qui permet à l'essaim d'explorer plus de territoire au fil du temps. Étant donné que l'essaim est composé de nombreux individus redondants, l'essaim lui-même est plus tolérant aux erreurs qu'un robot effectuant la même tâche, en ce sens qu'il peut perdre des individus (par exemple en raison de pannes matérielles) sans affecter les performances globales. Cela évite le problème de point de défaillance unique dans les systèmes robotiques à commande centrale.

Le problème de foraging est un benchmark dans la robotique en essaim. Il constitue une abstraction de plusieurs applications du monde réel à savoir la recherche de ressources dans des environnements potentiellement dangereux. Dans un problème de foraging, un ensemble de robots sont chargés de chercher, collecter et transporter des objets distribués de façon aléatoire à un au plusieurs dépôts.

L'objectif de ce chapitre est présenté le problème de foargin Multi-Robots, avec les travaux reliés. Nous présentons dans un premier temps une introduction de l'intelligence en essaim et la robotique en essaim, en présentant les différentes caractéristiques ainsi que les comportements de robotique en essaim, puis nous focalisons sur le comportement de foraging. Puis, nous présentons les travaux reliés aux problèmes de foraging dans un environnement avec un seul dépôt et environnement avec multi-dépôts.

2. Intelligence en essaim

2.1. Définitions

L'intelligence en essaim (ou *swarm intelligence* en anglais) a été introduite pour la première fois par Gerardo Beni et JingWang en [1989]. Le mot "*essaim*" évoque l'image d'un grand nombre de petits insectes où chaque individu effectue une tâche simple, mais dont l'action produit un comportement complexe dans son ensemble *[Kennedy et al., 2001]*. Les essaims sont définis comme des collections de nombreux individus simples qui interagissent avec les autres individus et l'environnement en utilisant un contrôle décentralisé et auto-organisée pour atteindre leurs objectifs *[Miller Peter, 2007]*.

La combinaison de leurs comportements simples ou microscopiques provoque des actions beaucoup plus complexes et macroscopiques, qui permettent à l'ensemble du système d'obtenir des résultats remarquables dans son ensemble.

Dorigo et Theraulaz dans *[Bonabeau et al. 1999a]* étend la définition de l'intelligence en essaim au-delà du contexte de la robotique d'origine :

L'intelligence en essaim inclut toute tentative de conception d'un algorithme ou d'un dispositif visant à résoudre des problèmes de façon distribuée, inspirée du comportement collectif des insectes sociaux ou d'autres sociétés animales.Le message est clair : les phénomènes biologiques sont la principale source d'inspiration pour l'intelligence en essaim. Ces auteurs décrivent de plus cette forme d'intelligence comme une "Intelligence collective émergeant d'un groupe d'agents simples."

La préface de Merkle et Blum dans l'ouvrage *[Blum et Merkle, 2008]* propose une définition pragmatique de l'intelligence en essaim du point de vue informatique :

L'intelligence en essaim est une discipline de l'intelligence artificielle moderne qui traite de la conception de systèmes multi-agents en vue d'applications telles que l'optimisation et la robotique.

Les avantages pressentis de l'essaim de robots par rapport à un système de robots à contrôle centralisé, découlent directement de ces propriétés *[Charrier, 2009]*:

- Des unités simples sont plus faciles à produire en masse et sont interchangeables ;
- Le système est plus fiable, plus robuste aux diverses pannes ou perturbations auxquelles il peut être soumis du fait de la redondance des unités ;
- Le système en essaim est doté d'importantes capacités d'adaptation à son environnement ;

- Enfin le système en essaim peut résoudre des problèmes compliqués du fait de ses multiples unités de calcul, bien au-delà des problèmes que peuvent résoudre les entités individuellement.

(a) (b) (c)

Figure1.1 – Exemples des systèmes d'intelligence en essaim :
(a) Troupeaux d'oiseaux (b) Groupes de poissons (c) colonies de fourmis

2.2. Propriétés d'un système d'intelligence en essaim

Par ses principes, l'intelligence en essaim essaye de résoudre certains problèmes liés à la robotique. Ce sont à la fois la mise à l'échelle des systèmes, une tolérance accrue aux erreurs et une flexibilité qui sont aux centres de celle-ci *[Mayeur, 2014]*.

- **Mise à l'échelle:** Le nombre de robots impliqués suppose que l'ajout ou le retrait de robots ne doit pas réduire de manière significative les résultats de l'ensemble. Cette mise à l'échelle doit également permettre de résoudre les problèmes de différentes tailles.
- **Tolérance aux erreurs**: Par la redondance du système, et l'absence de contrôle centralisé, le système est aussi robuste que possible. Si un robot devient défaillant, il peut être remplacé par un autre robot. L'utilisation de capteurs ne pouvant capter que des données locales peut également mener à de mauvaises décisions lors de l'exécution des différents algorithmes. Toutefois, la redondance du système mène à ce que les erreurs se compensent, et même si un robot venait à prendre une mauvaise décision, la majorité des robots serait-elle même capable de réagir convenablement pour résoudre le problème.
- **Flexibilité:** C'est la possibilité pour un essaim de robots de résoudre différentes tâches dans différents environnements. Ceci s'inscrit dans les conséquences de la redondance des systèmes. Si les mêmes robots sont utilisables pour différentes tâches et environnements, ils peuvent alors s'adapter à ceux-ci.

3. Robotique en essaim

3.1. Définitions

L'utilisation de contrôle centralisé et de communication globale dans des système complexe devient infaisable et il faut recourir à d'autres approches. La robotique en essaim (swarm robotics en anglais) est une nouvelle approche à la coordination distribuée des plusieurs robots qui s'inspire de l'intelligence en essaim.

La Robotique en essaim selon Dorigo, Birattari et Brambilla[Dorigo and Gambardella, 1997] est l'étude de la manière de créer des groupes de robots qui s'exécutent sans dépendre d'aucune infrastructure extérieure ou aucune forme de contrôle centralisée. Dans un essaim de robots, le comportement collectif des robots résulte de l'interaction locale entre les robots et l'environnement dans lequel ils interagissent. Les systèmes de robotique en essaim sont des systèmes tolérants aux erreurs, extensible et flexible.

C'est une approche prometteuse lorsque différentes activités doivent être effectuées simultanément, lorsque la redondance élevée et l'absence d'un point de défaillance unique sont souhaitées, et quand il est techniquement impossible de mettre en place l'infrastructure nécessaire pour contrôler les robots de façon centralisée [Zedadra, 2015].

Foot-bot *EPuck* *Drones*

Figure 1.2: Exemples de systèmes de la robotique en essaim

3.2. Principes de robotique en essaim

Dans ce qui suit, nous présentons plusieurs caractéristiques envisagées par un essaim de robots *[Mayeur, 2014]*.

- **Grand nombre d'individus :** Les essaims se construisent autour de nombreux robots semblables qui interagissent afin de résoudre le problème posé. Ces essaims peuvent avoir des tailles variables en fonction du problème étudié : le nombre d'individus

optimal n'est pas toujours simple à déterminer et peut nécessiter une étude approfondie.

- **Contrôle non centralisé et perception locale:** Par la taille et les limitations physiques des robots, qui sont supposés être les plus simples possible, il est supposé que ceux-ci ne possèdent qu'une partie des informations concernant leur environnement. Ainsi, chaque robot est livré aux données qu'il peut récolter et éventuellement aux communications locales entre les robots.
- **Redondance** Dans les essaims de robots, il est souhaité de posséder une certaine homogénéité entre les individus. Il est, en effet, sous-entendu que les robots sont interchangeables et qu'une fois inter changés, ils réagissent de manière similaire. Cette homogénéité implique une certaine redondance au niveau des individus, ce qui augmente grandement l'extensibilité ainsi que la flexibilité du système.

3.3. Comportements de la robotique en essaim

Nemriniet Martinoli *[Nembrini and Martinoli, 2007]* ont classifié les comportements de la robotique en essaim en deux catégories : Comportements de base et comportements combinés :

A. Comportements de base

Les comportements simples sont à la base de construction des comportements plus complexes. Quatre de ces comportements ont été traités de manière systématique par la communauté : dispersion, agrégation, mouvement collectif et décision collective.

1) ***Dispersion:*** La dispersion contrôlée dans un environnement est une composante comportementale importante pour un essaim de robot.
2) ***Agrégation:*** L'objectif principal de ce problème est de regrouper tous les robots d'un essaim dans une région de l'environnement par exemple pour échanger des informations.
3) ***Mouvement collectif:*** Le mouvement collectif est un autre comportement de base important. Du fait des limitations des dispositifs actuels de positionnement relatif embarqué.
4) ***Choix collectif:*** La possibilité pour un essaim de robot de récolter différentes observations de l'environnement en des localisations différentes et de les partager, permet d'augmenter la capacité de prise de décision des individus.

B. Comportements combinés :

Certains comportements combinés et plus complexes ont été identifiés pouvant être directement implémentés dans une variété d'applications ou pour des tâches bien spécifiques.

1) ***Exploration:*** L'exploration implique la nécessité pour un essaim de robots de découvrir l'environnement. Ce comportement est le plus étudiée dans le domaine de robotique en essaim.
2) ***Couverture:*** La couverture d'une région implique un passage systématique en tous les points de région.
3) ***Localisation de cible:*** Cette tâche peut avoir de nombreuses variantes, dépendantes de la nature de la cible et de la manière de la localiser. Bien que beaucoup de recherche utilisant des modèles numériques ou analytique se préoccupent de la localisation de cibles.
4) ***Manipulation collective:*** Les exemples de manipulation collective se présentent soit sous la forme d'un transport collectif d'un unique objet, soit sous la forme d'une agrégation d'objets pour une ségrégation.
5) ***Fourragement:*** Analogue à la recherche de cible, la tâche de fourragement, directement inspirée des fourmis, y ajoute la nécessité de ramener les cibles en un point.

3.4. Application de la robotique en essaim dans le monde réel:

YazdaniHasen [Yazdani, 2017] a cité les différentes applications de la robotique en essaim dans le monde réel :

- ***Missions de secours :*** Des essaims de robots pourraient être envoyés dans desendroits où les secours ne pourraient pas être atteints, ce qui sauverait des vies.
- ***Tâches minières :***Un essaim pourrait couvrir un champ de mines et les robots seraient sacrifiés sur les mines. Les robots Swarms peuvent être utilisés dans l'armée pour former une armée autonome.
- ***Surveillance autonome et la surveillance de l'environnement :*** afin d'étudier les paramètres environnementaux, de rechercher les survivants et de localiser les sources de dangers telles que déversements de produits chimiques ou de gaz, pollution toxique, fuites de canalisations, radioactivité.

- ***Couverture d'une vaste région :*** Les robots peuvent se disperser et effectuer des tâches de surveillance, par exemple dans des forêts. Cela peut être utile pour détecter des événements dangereux, comme une fuite d'une substance chimique. Dans les applications militaires.
- ***Agriculture :***Un scénario de surveillance/ cartographie décentralisé et la mise en œuvre d'un cas d'utilisation pour la détection et la cartographie des mauvaises herbes dans un champ par un groupe de petites véhicules aériens sans pilote.
- ***Nettoyage des déchets toxiques :*** la recherche et le sauvetage et la collecte d'échantillons de terrain font partie des applications importantes des robots en essaim.
- ***Exploration et la cartographie :*** permettrait de gagner du temps et de l'argent.
- ***Domaine médical :*** l'utilisation de nano-robots se déplaçant dans les veines et les artères humaines (par exemple, pour lutter contre certains types de cancer).
- ***Tâches dangereuses :*** De nombreuses industries utilisent des objets dangereux comme des brûleurs, des produits chimiques, la fabrication d'armes nucléaires, etc. Dans ce cas, l'utilisation d'essaims peut réduire le danger dans de tels types d'industries.

4. Problème de foraging

4.1. Aperçu générale

Le foraging est un problème de référence, particulièrement pour les systèmes multi-robots. C’est un problème très important pour différentes raisons:

1. Les inspirations fournissent de la compréhension du comportement de recherche de nourritures observé chez les insectes sociaux ;
2. Inclut la coordination de plusieurs tâches, comme l’exploration, la collection physique des objets, transport des objets, la navigation ou le retour au dépôt, et le dépôt de l’objet dans la base ;
3. Le foraging requière de la coopération entre les agents, soit par communication de l’endroit de la source aux autres agents et/ou le transport coopératif des objets qui sont beaucoup plus difficiles à transporter par un seul agent.

Dans une mission de foraging, un essaim de robots doit résoudre collectivement plusieurs sous-tâches. Tout d’abord, l’essaim doit quitter la station de base et explorer

l'environnement à la recherche de ressources. Une fois découverts, les robots de l'essaim doivent ramasser la nourriture et retrouver le dépôt pour la déposer, une fois le dépôt est atteint, le robot dépose l'objet er résume sa recherche. L'état par défaut dans le foraging c'est l'exploration. Les individus dans un essaim peuvent effectuer des interactions entre eux, ce qui peut affecter positivement ou négativement les performances globales *[Ericksen et al., 2017]*.

4.2. Types de Foraging

Les objets sont transportés vers un ou plusieurs dépôts. Dans le premier cas, on parle du foraging centralisé (CPF) et dans le deuxième on parle du foraging multi-dépots (MPF).

1) **Le foraging centralisé :** Le foraging centralisé (Central-Place Foraging 'CPF') est une tâche collective canonique communément étudiée en robotique en essaim *[Brambilla et al., 2013]*. Les robots partent d'un dépôt situé au centre de leur environnement pour rechercher des ressources et reviennent à ce dépôt central pour déposer les ressources collectées.

   A cause de la surpopulation au tour du dépôt central, les collisions augmentent avec le nombre de robots. Par conséquent, un seul dépôt central ne peut pas desservir efficacement un grand nombre de robots. De plus, les ressources situées loin du dépôt central imposent de longs temps de trajet.
2) **Foraging multi-dépôts:** On s'attend à des rendements décroissants pour la recherche de nourriture dans les lieux centraux, car les robots de plus gros essaims se déplacent en moyenne plus loin pour collecter plus de cibles, et il y a plus de collisions avec plus de robots. Le foraging multi-dépots (Multi-Place Foraging 'MPF') est inspiré par les comportements observés chez des groupes d'insectes et de primates, ainsi que par le système immunitaire. Par exemple, les colonies de fourmis argentines sont composées de plusieurs nids s'étendant sur des centaines de mètres carrés *[Flanagan et al.2013]*. Aussi, les communautés de singes-araignées sont considérées comme des systèmes MPF, du fait que les singes sélectionnent un site de couchage à proximité des zones de nourriture actuelles, la stratégie MPF entraînant des coûts de déplacement les plus bas *[Chapman et al., 1989]*. Même la nature décentralisée et sous-modulaire du système immunitaire augmente l'efficacité de la recherche de nourriture des cellules immunitaires qui s'agrègent dans les ganglions lymphatiques répartis dans tout le corps *[Banerjee and Moses, 2010b]*.

Deux instances du foraging existent, que ce soit dans les CPF ou les MPF :

1) **Foraging durable (sustainable) :** est un foraging dans lequel un ou plusieurs robots doivent garder le système vivant en récupérant des ressources dans l'environnement et le déposer dans un ou plusieurs dépôts, ces ressources sont extraites par un système externe, donc l'objectif des robots c'est de récolter un maximum possible d'objets pour assurer une survie de leur dépôt.
2) **Foraging non-durable (non-sustainable) :** les robots doivent collecter des ressources dans l'environnement et de les déposer dans un ou plusieurs dépôts, il n'existe aucun système externe qui exploite ces ressources collectées.

5. Foraging Centralisé : Travaux reliés

1) Central Place Foraging Algorithm (CPFA) *[Hecker and Moses, 2015]*

Hecker et Moses *[Hecker and Moses, 2015]* ont proposé l'algorithme CPFA (Central Place Foraging Algorithm), conçu pour imiter les comportements des fourmis Arracheuse de graines qui régissent la mémoire, la communication et les mouvements.

Les robots se dispersent initialement vers des emplacements aléatoires, puis ils utilisent une marche aléatoire corrélée non informée pour localiser les ressources. Les robots collectent une ressource à la fois et y retournent ou abandonnent la recherche si aucune ressource n'a été localisé pendant une certaine période de temps. Si une ressource est localisée, le robot enregistre le nombre de ressources détectées (densité de ressources dans la région locale) dans le voisinage à 8 cellules de la ressource trouvée. Un robot individuel peut se souvenir de l'emplacement d'une ressource trouvée précédemment et revenir à plusieurs reprises au même emplacement à l'aide d'un processus appelé fidélité de site. Les robots peuvent également communiquer en utilisant des phéromones pour recruter des robots dans des groupes de ressources connus. Le Figure 1.3 décrit schématiquement le comportement des robots.

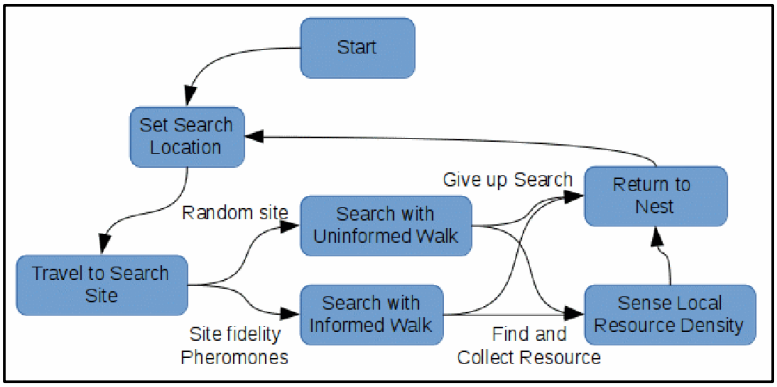


Figure 1.3: La machine d'état de comportement CPFA*[Hecker and Moses, 2015]*.

Le CPFA a été implémenté dans des environnements physiques réels utilisant un dépôt central illuminé par une balise que les robots peuvent détecter. Les robots utilisent une combinaison de distance ultrasonore, de cap boussole magnétique, d'odométrie temporelle et d'une caméra embarquée faisant face à l'avant pour estimer l'emplacement des points de passage de phéromone et lieux de retour via Fidélité de site *[Hecker and Moses, 2015]*.

2) Distributed Deterministic Spiral Search Algorithm (DDSA)*[Matthew et al, 2016]*

L'algorithme de recherche en spirale déterministe distribuée (Distributed Deterministic Spiral Search Algorithm- DDSA), vise à éliminer le caractère aléatoire inhérent aux approches aléatoires en faisant en sorte que les agents se déploient vers l'extérieur à partir du dépôt de collecte selon un modèle spiral. Les agents DDSA rapportent les ressources au dépôt dès qu'elles ont été trouvées, puis retournent toujours à l'emplacement où ils ont trouvé la dernière cible. C'est pour que la spirale de recherche puisse être rejointe au moment où elle a été interrompue, Le processus de retour à l'emplacement de la dernière cible trouvée s'appelle fidélité de site et constitue une stratégie de recherche courante chez les fourmis.

DDSA exige que chaque robot connaisse la taille de l'essaim et son index, les robots s'éloignent de l'emplacement central et effectuer un mouvement Nord (N), Est (E), Sud (S), Ouest (W), ces mouvements son symétrique c'est-à-dire $D_N=D_S$ et $D_E=D_W$(oùD_H est la distance parcourue par le robot actuel dans une direction cardinale donnée, H pour un nombre de circuits particulier, c). Le nombre de circuits est le nombre de fois qu'un robot a effectué un mouvement dans les quatre directions.

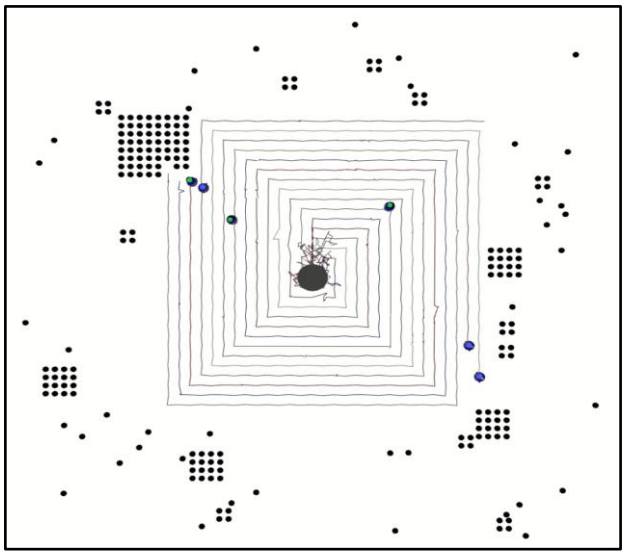


Figure 1.4: DDSA s'exécutant dans ARGoS, vue de dessus.*[Matthew et al, 2016]*

DDSA a montré de meilleures performances par rapport au CPFA en termes de temps de recherche et collecte, grâce à la collecte incrémentale des objets. Cependant, le CPFA a surperformé le DDSA en termes de nombre d'objets collectés dans le temps pour les gros essaims avec plus de 15 robots, parce que DDSA souffert de collisions de robots dans des environnements plus encombrés.

3) *NEAT Foraging Algorithm (NeatFA):* *[Ericksen et al., 2017]*

L'algorithme NeatFA se base sur les contrôleurs de réseaux deneurones qui peuvent être générés automatiquement pour la robotique en essaim dans le problème de foraging à l'aide de Neuroevolution Augmented topologies (NEAT)*[Stanley and Millkkulainen, 2002]*.

La NEAT *[Stanley and Millkkulainen, 2002]* est un algorithme génétique (AG) spécialisée dans la conception de réseaux de neurones. Elle recherche une bonne conception en encodant un réseau minimal dans un chromosome pour la population initiale, puis en faisant évoluer le réseau en utilisant une croissance de réseau prudente, des croisements basés sur les fonctions et la spéciation. Les réseaux construits par NEAT sont constitués d'un graphe orienté de perceptrons connectés. Chaque perceptron combine les entrées réelles des capteurs du système et des autres perceptrons pour produire des sorties de signal de contrôle.

NeatFA permet au réseau de neurones de déposer une goutte de phéromone si un neurone de sortie désigné a une valeur supérieure à 0 et qu'il n'y a pas de phéromone dans le rayon R = 5 cm. En outre, le robot peut détecter la présence d'une goutte de phéromone dans le rayon 2R (une valeur +1 au niveau d'un neurone d'entrée désigné et -1 en l'absence de phéromone). NeatFA le met en œuvre en incluant un ensemble de détecteurs de distance répartis uniformément dans le corps de chaque robot. Les valeurs du détecteur de distance sont des entrées brutes du réseau de neurones.

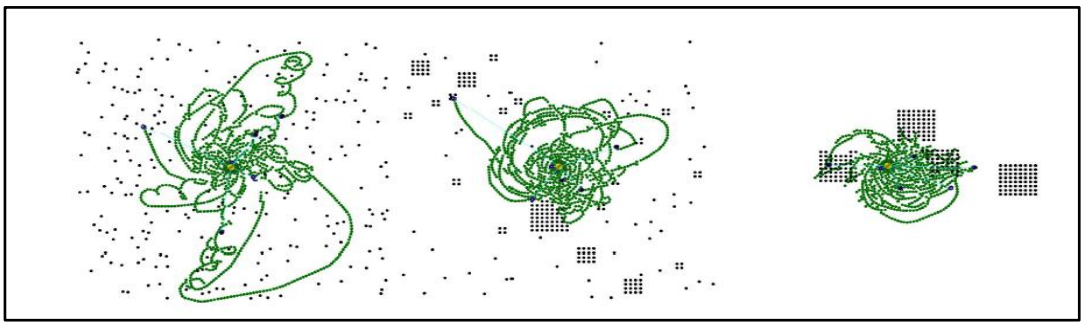


Figure1.5:NeatFA fonctionnant dans les trois distributions ; de droite à gauche: aléatoire, semiclusterisé et en cluster. *[Ericksen et al., 2017]*

Comme dans la CPFA et la DDSA, les contrôleurs évolués savent où se trouve le nid dans l'arène pour faciliter la navigation vers le nid.

Dans un environnement semi-cluster avec différentes tailles d'essaims, la performance de NeatFA est similaire à celle de CPFA et meilleur que DDSA, ce qui montre que NeatFA évite le problème d'encombrement des nids observé dans DDSA.

4) ***Multimodal Central Place Foraging (MMDF)****[Dolan-Stern et al, 2018]*

Multimodal Central Place Foraging (MMDF) est inspirée par la façon dont les colonies d'abeilles divisent la recherche de nectar en plusieurs rôles comportementaux. MMDF est un algorithme qui utilise plusieurs rôles pour organiser la recherche et rassembler le comportement, diviser l'espace de recherche entre les agents afin de réduire les collisions, et partager la recherche et le suivi des progrès entre les agents. De manière similaire à DDSA, MMDF utilise un modèle de recherche en spiral en raison de sa couverture complète, de son évolutivité et de son chevauchement minimal. MMDF écarte de DDSA en ce sens que les agents ne collectent aucune ressource jusqu'à ce que l'espace de recherche soit complètement couvert et que l'emplacement de toutes les ressources sur le chemin de recherche soit déterminé. Une fois que la zone de recherche est couverte, en affectant des agents à la collecte des ressources les plus proches d'eux dans des zones séparées les unes des autres.

L'état de chaque cible de l'environnement est partagé entre chaque agent du système. Chaque cible est initialisée pour être dans l'état inconnu, ce qui signifie que sa position ne peut pas encore être déterminée. Quand l'agent détecte la cible, il envoie un message comprenant l'identificateur unique de la cible ainsi que la position à laquelle elle a été vue. À la réception du message de détection, chaque agent met à jour sa liste d'états cibles locale afin que la position de cette cible soit enregistrée et que son état soit défini sur Détecté.

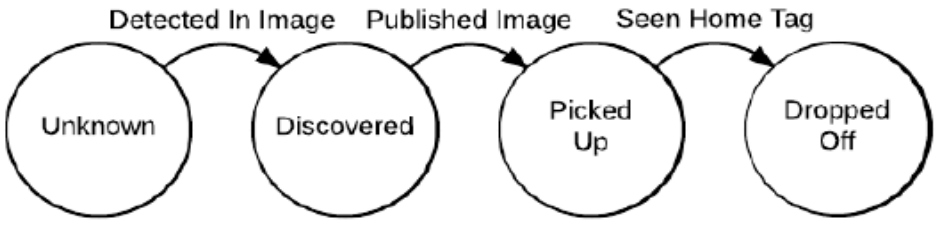


Figure 1.6: Diagramme d'état pour les cibles.

A la fin de la mission de recherche des emplacements des cibles, les collecteurs retournent dans chacun des emplacements cibles afin de les récupérer et de les amener au nid.

En route vers l'emplacement prévu de la cible réclamée, toute cible inconnue ou découverte rencontrée sera récupérée à la place. Si le collecteur est arrivé à l'emplacement prévu d'une cible et que rien n'a été détecté, il effectuera une recherche aléatoire pour tenter de trouver la cible. Une fois la cible localisée, le collecteur diffusera un message de collecte contenant l'identifiant unique de la cible. Chaque agent récepteur met à jour son état cible de sorte qu'aucune autre tentative de collecte de cette cible ne soit effectuée. Une fois ce message publié, l'agent collecteur retourne maintenant au nid.

Une fois que le collecteur a atteint l'emplacement de nid, il recherche des indices visuels du nid. Si le nid de base n'est pas visible, le collecteur effectuera une recherche aléatoire sur cet emplacement. Une fois que le nid est visible, le collecteur lâche la cible et diffuse un message signalant aux autres agents que cette cible a été collectée. Le secteur revendiqué par cet agent est libéré lorsque le message "Dropoff" correspondant a été reçu par les autres agents. Le processus de collecte de l'agent déposant la cible est ensuite répété.

Les résultats montrent qu'au bout de 15 minutes, DDSA recueille clairement un pourcentage élevé de cibles par rapport au MMDF.

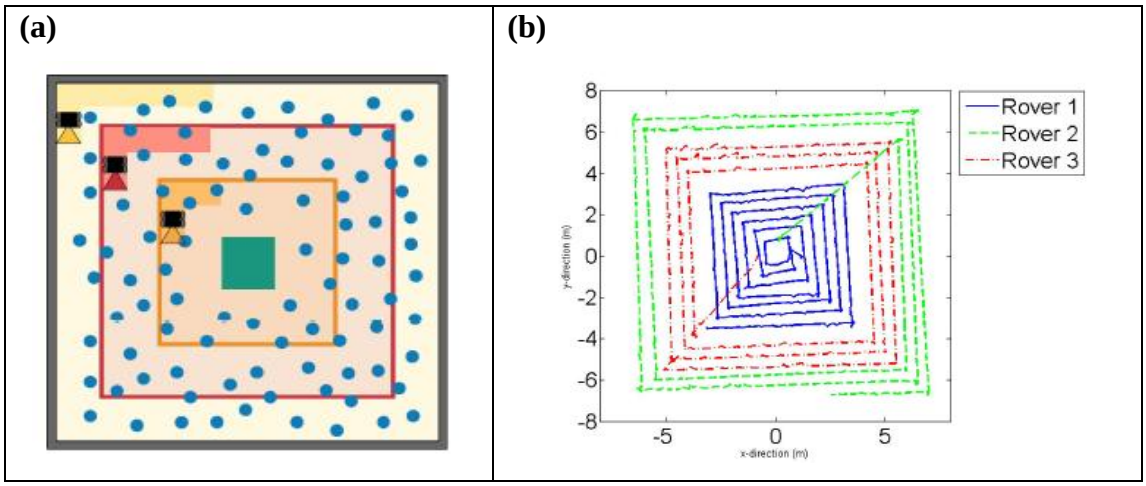


Figure1.7 : (a) Visualisation du comportement de recherche, (b) Tracé de l'emplacement des agents pendant les 350 premières secondes d'exécution de l'algorithme MMDF. *[Dolan-Stern et al, 2018]*

6. ForagingMulti-Dépôts : Travaux reliés

1) *Multiple Place Foraging Algorithm (MPFA$_{static}$)* *[Lu et al., 2016]*

Dans la MPFA$_{static}$, les robots sont répartis uniformément autour des nids. Les robots partent d'un nid aléatoire, mais retournent au nid le plus proche à leurs positions après avoir trouvé une ressource. Les robots ont à priori connaissance de l'emplacement des nids. L'utilisation

de plusieurs points de collecte est la différence fondamentale entre le CPFA et le MPFA; tous les autres composants des deux algorithmes de foraging sont maintenus délibérément identiques afin de tester l'effet de plusieurs nids sur l'efficacité de la recherche par essaim. Comme dans la CPFA, les robots utilisent la fidélité au site ou suivent phéromones pour exploiter les zones riches en ressources.

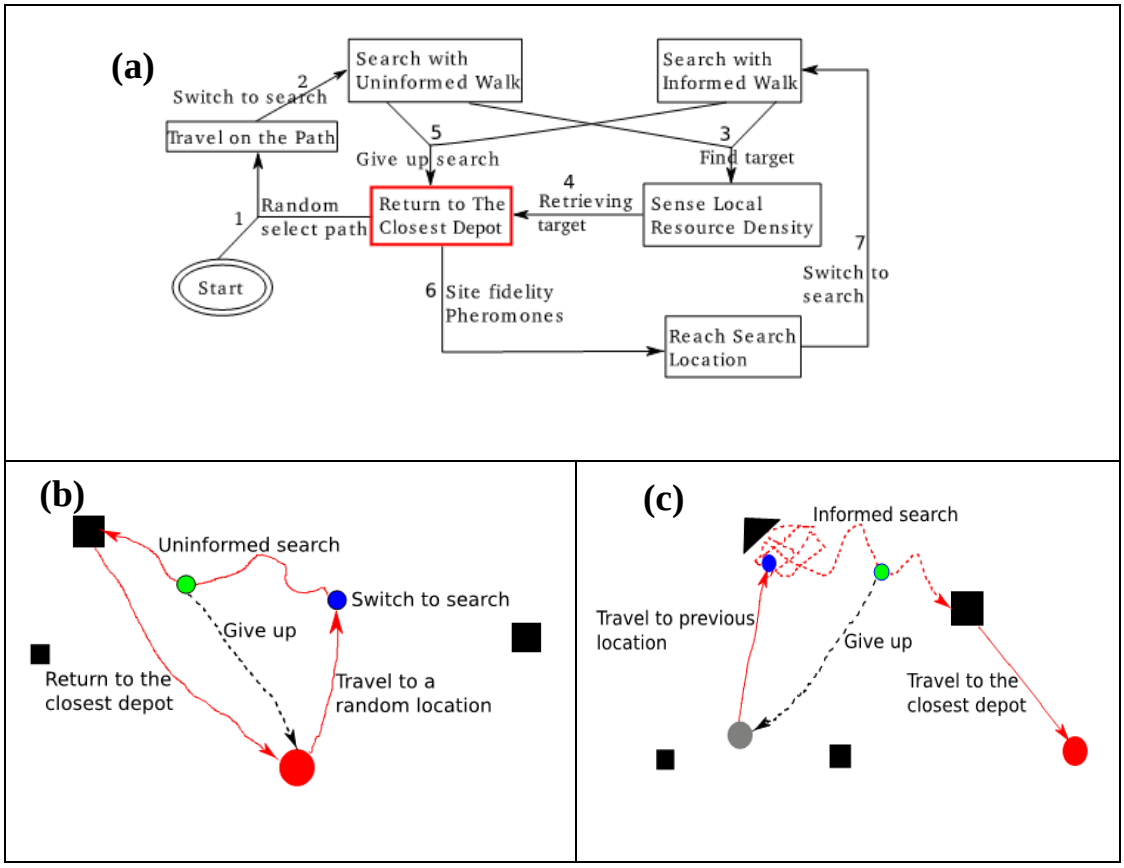


Figure 1.8: (a) Comportement des robots MPFA, (b) Un seul cycle de recherche non informée et (c) Un seul cycle de recherche éclairée.*[Lu et al., 2016]*

Dans la MPFA, les robots effectuent des recherches globales, comme dans la CPFA, ils peuvent voyager dans l'arène entière. La principale différence est que les robots retourneront toujours dans le nid le plus proche de l'emplacement où ils ont découvert les ressources. Ils partagent les points de cheminement aux phéromones localement à leur nid actuel. Cela contraste avec le CPFA, où les points de cheminement aux phéromones sont associés au nid placé au centre et sont disponibles globalement pour tous les robots. Étant donné que les robots retournent toujours au nid le plus proche avec une ressource trouvée dans la MPFA, les informations détectées pertinentes pour un voisinage de ressource donné sont toujours associées au nid le plus proche de la position du voisinage identifié. Ainsi, si un robot suit un point de cheminement de phéromone depuis un nid, la distance entre le nid et la destination

de la phéromone est la distance la plus courte jusqu'au voisinage de la ressource identifiée par le point de cheminement.

2) ***Multiple-place swarm foraging with dynamic depots (MPFA$_{dynamic}$)*** *[Lu et al., 2018]*

L'algorithme de foraging Multi-dépôts dynamiques (*MPFA$_{dynamic}$*), est une extension de l'algorithme MPFA [lu et al., 2016] dans lequel les dépôts se déplacent vers de nouveaux emplacements en fonction des emplacements des cibles détectées par les robots. L'utilisation de dépôts mobiles constitue la différence fondamentale entre *MPFA$_{static}$* et *MPFA$_{dynamic}$*, tous les autres composants des deux algorithmes de foraging sont maintenus délibérément identiques. Comme dans *MPFA$_{static}$*, les dépôts sont initialement répartis uniformément dans l'environnement, et les robots sont répartis de manière uniforme dans chaque dépôt. Les dépôts se déplacent vers de nouveaux emplacements en fonction des informations de position des cibles détectées par les robots, l'algorithme de regroupement *k-means++[Arthur and Vassilvitskii, 2007]* calculera l'emplacement des dépôts afin de minimiser la distance de parcours requise pour collecter toutes les cibles.

La *Figure 1.9 (a)*montre un exemple de dépôt alloué de manière dynamique, dans lequel six piles de cibles sont classées en quatre groupes et quatre dépôts sont placés au centre de la grappe. La *Figure 1.9 (b)*montre comment un dépôt se déplace en fonction des informations de position détectées des cibles rapportées par les robots. Un dépôt (cercle gris) est situé au centre de gravité *C1* des cibles détectées (carrés bleu foncé) aux positions *p1, p2* et *p3,* où *w1, w2* et *w3* sont respectivement le nombre de cibles détectées par les robots à chaque position. Après un certain temps, si les cibles en position *p1* sont complètement collectées par des robots, les points de cheminement aux phéromones en *p1* se désintègrent. Si, en même temps, les cibles *w4* sont détectées à un nouvel emplacement *p4*, le dépôt se déplacera ensuite vers le centre de gravité *2* des cibles détectées (cercle rouge) aux positions *p2, p3* et *p4*.

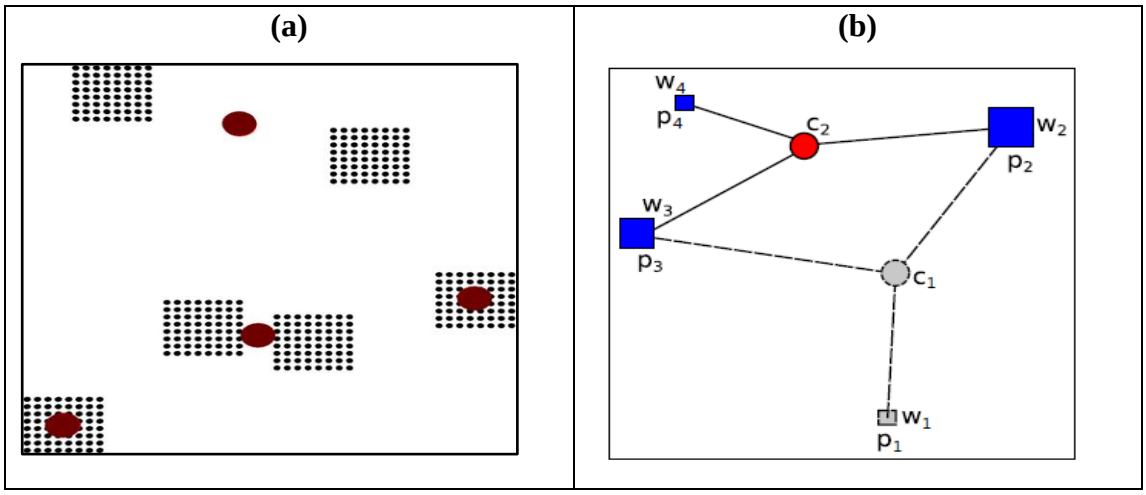


Figure 1.9: (a) Un exemple de dépôt alloué dynamiquement utilisant l'algorithme de classification kmeans ++. (b) Mouvement de dépôt dans $MPFA_{dynamic}$.*[Lu et al., 2018]*

Lu et al. *[Lu et al., 2018]* ont testé 4 variantes de l'algorithme d'alimentation multiple et un algorithme d'alimentation central (CPFA). Parce que ces algorithmes inspirés par les fourmis sont conçus pour collecter rapidement des cibles plutôt que pour une collection complète de toutes les cibles, le temps nécessaire pour collecter 88% des cibles disponibles dans chaque expérience. Dans le premier ensemble d'expériences avec 24 robots dans une arène de 10 × 10 m, la durée moyenne de recherche de $MPFA_{dynamic}$ sur les trois distributions des cibles est 41% plus rapide que le CPFA centralisé et 13% plus rapide que $MPFA_{static}$.

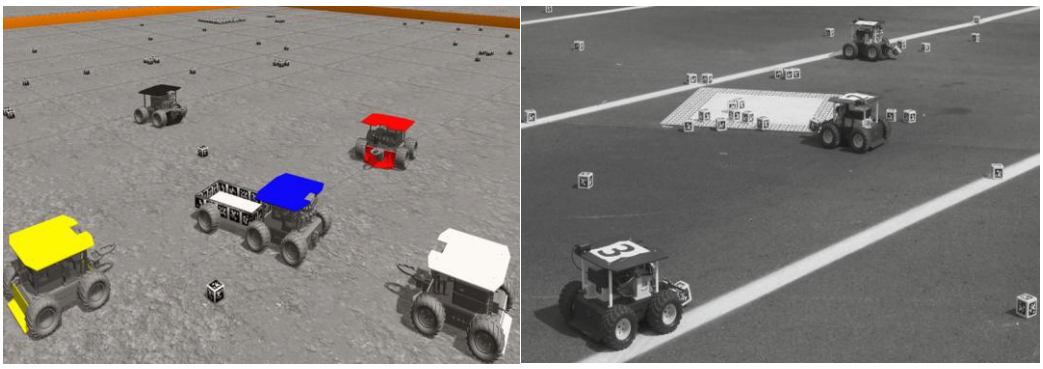


Figure 1.10: (a) Un dépôt mobile à couverture bleue et quatre robots chercheurs simulés dans Gazebo, (b) Un essaim de 6 robots (3 représentés) mettant en place un dépôt central dans une arène de 23 × 23 m *[Lu et al., 2018]*

En plus d'avoir des temps de recherche plus rapides pour toutes les tailles d'arène et toutes les distributions cibles, $MPFAd_{ynamic}$ est également plus évolutif que CPFA et $MPFA_{static}$. Les temps de recherche de nourriture $MPFA_{dynamic}$ sont particulièrement rapides pour les grandes arènes et les cibles en grappes.

Ainsi, en utilisant des dépôts mobiles qui s'adaptent aux conditions locales, $MPFA_{dynamic}$ est une solution efficace et évolutive qui minimise le goulot d'étranglement de la CPFA au centre de l'emplacement et améliore les durées de recherche de nourriture par rapport à $MPFA_{static}$ sans nécessiter d'informations globales.

7. Comparaison qualitative des travaux reliés

Foraging	Référence	Algorithme	Commu-entre robots	Distribution des Objets	Stratégie d'exploration	Stratégie de Homing	Basé Intelligence en essaim	Mesures de performance	Durabilité de nid
Foraging à dépôt Central (CPF)	*[Hecker and Moses, 2015]*	CPFA	Indirecte (phéromones)	Aléatoire, partiellement, entièrement groupés.	Marche aléatoire uniformément (ou informée) corrélée	Position de dépôt	Oui (Fourmis + AG)	Temps de collision, de recherche et de voyage	Non
	[Matthew et al, 2016]	DDSA	Indirecte (phéromones)	Aléatoire, partiellement, entièrement groupés.	Marche en spéral	Suivre la lumière émise par le dépôt	Oui (Fourmis)	Temps de collision, de recherche et de voyage	Non
	[Ericksen et al., 2017]	NeatFA	Indirecte (phéromones)	Aléatoire, partiellement, entièrement groupés.	Marche aléatoire uniformément (ou informée) corrélée	Suivre la lumière émise par le dépôt	Oui (Fourmis + NEAT)	Temps de collision, de recherche et de voyage	Non
	[Dolan-Stern et al, 2018]	MMDF	Directe (envoie des méssages)	Aléatoire, entièrement groupés.	Marche en spéral	Suivre la lumière émise par le dépôt	Oui colonies d'abeilles	Temps de collision, de recherche et de voyage	Non
Foraging à Multi-Dépôts (MPF)	*[lu et al., 2016]*	$MPFA_{static}$	Indirecte (phéromones)	Aléatoire, partiellement, entièrement groupés.	Marche aléatoire uniformément (ou informée) corrélée	Position des dépôts	Oui (Fourmis + AG)	Temps de collision, de recherche et de voyage	Non
	[Lu et al., 2018]	$MPFA_{dynamique}$	Indirecte (phéromones)	Aléatoire, partiellement, entièrement groupés.	Marche aléatoire uniformément (ou informée) corrélée	Position des dépôts	Oui (Fourmis + AG)	Temps de collision, de recherche et de voyage	Non

Tableau 1.1 : Comparaison qualitative des travaux reliés.

Discussion:

Le foragingCPF donne de bons résultats dans des environnements simples, les rendements globaux de l'essaim diminuent lorsque la taille du scénario ou le nombre de robots augmente. Ce type de foraging pose des problèmes dans plusieurs situations :

- Lorsque la zone de recherche augmente, les robots ne disposent pas de la capacité nécessaire pour communiquer et calculer efficacement les données, ce qui nuit à l'échelle du système ;
- Les robots d'essaims doivent en moyenne voyager plus loin, ce qui réduit le temps de trajet ;
- Comme il faut plus de robots pour couvrir la zone d'intérêt, le nombre de collisions à proximité du nid central augmente, ce qui engendre l'obstruction et la surpopulation des robots ;
- Les systèmes basés sur CPF ne conviennent pas à un déploiement dans des zones plus vastes et plus dynamiques.

Pour corriger ces défauts, les chercheurs sont dirigés vers le foraging multi-dépôts (MPF) :

- Il repose sur plusieurs nids plutôt que sur un nid central. Les nids sont dispersés dans toute la région et chaque robot situé à côté de l'essaim peut modifier son nid correspondant en fonction de son emplacement et de son statut énergétique ;
- Les dépôts ne sont pas toujours statiques, ce qui rend l'algorithme MPF plus adapté aux environnements réels tels que les zones urbaines, car les essaims peuvent s'adapter à une plus grande variété de situations.

Depuis le *Tableau 1.1*, nous avons fait plusieurs constations :

- La majorité des travaux ont utilisé la marche aléatoire pour l'exploration ;
- La majorité des travaux utilisent les coordonnées spatiales dans la stratégie de homing pour faciliter la localisation du dépôt et gagné du temps, certains d'autres utilisent la lumière ;
- Quelques travaux utilisent les algorithmes bio-inspirés pour améliorer l'exploration et imiter le comportement de communication indirecte non couteux des insectes sociaux ;

8. Conclusion

Dans ce chapitre nous avons présenté le principe général de l'intelligence en essaim et leurs propriétés principales. Nous avons également présenté comment l'intelligence en essaim résout le problème de foraging. Plusieurs études sur l'attribution de tâches à des essaims de robots ont utilisé des approches inspirées de la biologie pour déployer des essaims homogènes de robots sur plusieurs sites. Nous avons fait une analyse et comparaison des travaux de foraging CPF, MPF soit durables ou non.

Dans le chapitre suivant, nous passerons à une proposition pour les problèmes de foraging durable, toute en inspirons des algorithmes issus de l'intelligence en essaim.

CHAPITRE 2

CHAPITRE 2

Conception

1. Introduction

Le foraging classique se trouve divisé en deux catégories : CPF et MPF. Dans les deux catégories, les robots sont chargés de chercher, collecter et transporter un ou plusieurs objets à un (CPF) ou plusieurs dépôts (MPF). Les travaux de foraging classique restent loin d'être appliqué en réalité, du fait que, ces derniers ne considèrent pas certaines caractéristiques nécessaires pour répondre aux exigences du monde réel tel que : l'adaptation en ligne, la durabilité du système et l'adaptabilité dans des scénarios critiques.

Une nouvelle extension du foraging classique nommée foraging durable (sustainable foraging), dans laquelle les robots tentent de répondre à des exigences des systèmes externes qui consomment les objets déposés aux dépôts à différentes reprises.

Nous cherchons dans ce travail, à proposer un algorithme de foraging multi-dépôts durable qui prend en considération les demandes qui lui provient de l'extérieure. Nous avons concentré sur la stratégie d'exploration basé sur l'algorithme de l'abeille, et sur un modèle d'interaction simple permettant de recruter des robots en cas de nécessité.

Nous commencerons par présenter la problématique, voire même la question de recherche qui fait le centre de ce travail. Par la suite, nous présentons les objectifs attendus et la proposition faite. Ensuite, nous présentons les deux algorithmes reliés à notre travail : Lévy Walk et l'algorithme de l'abeille.Nous présenterons aussi notre proposition qui est un système de foraging durable adaptatif dans un environnement multi-dépôts. Nous présentons ainsi, le comportement des robots et le pseudo code de l'algorithme proposé. Nous terminerons le chapitre par une conclusion.

2. Problématique

Les travaux de foraging classique ont tenté de proposer des solutions qui peuvent êtres envisageables sur plusieurs métaphores du monde réel tel que : le nettoyage, le déminage, l'exploration des environnements hostiles, la recherche et le sauvetage. Malheureusement, les travaux de foraging antérieurs ne tiennent pas compte de la nécessité de l'adaptation en

ligne du système, la durabilité du système et l'adaptabilité dans des scénarios critiques. Alors que ces caractéristiques sont cruciales pour que les systèmes de foraging soient déployable sur le monde réel.

L'objectif principal de ce type de problème est d'assurer une politique d'équilibrage de quantités des objets dans les défiants dépôts. On veut traiter la durabilité d'un système de foraging toute en focalisons à la fois sur la stratégie d'exploration et le modèle d'interaction.

Nous cherchons donc à répondre aux questions suivantes :

1. Est-il possible d'augmenter la survivabilité des dépôts en répondant à des demandes qui proviennent de l'extérieure à travers la stratégie d'exploration ?
2. Est-ce qu'une collaboration simple entre robots appartenant à des dépôts étrangers peut garantir la survie d'un dépôt en état d'alerte ?

3. Objectifs et Proposition:

Dans l'objectif de maintenir la durabilité dans un système de foraging multi-dépôts et pour répondre aux problématiques de recherche précédents, nous fixons les objectifs suivants :

1. Proposer un système de foraging multi-dépôt durable, qui arrive à maintenir sa stabilité pour des périodes de temps indéterminés ;
2. Utiliser une stratégie d'exploration stratégique, pour tester si le gain on objets peut augmenter la survie, voire la durabilité des dépôts et donc du système;

Nous avons donc:

1. Proposé un système de foraging multi-dépôt qui permet de satisfaire les demandes qui proviennent à ses différents dépôts dans des périodes de temps indéfinies. Les robots dans ce système utilisent un algorithme bio-inspiré basé algorithme de l'abeille pour l'exploration et la recherche des objets;
2. Si un manque a été détecté dans l'un des dépôts par l'un de ses robots, ce dernier doit recruter assez de robots appartenant à des dépôts étrangers pour collaborer à maintenir la survivabilité du dépôt infecté et donc de maintenir la survie globale du système.

4. Algorithmes reliés

Les deux algorithmes que nous avons utilisé dans notre système de foraging sont : Lévy Walk et l'algorithme de l'abeille. Nous donnerons dans cette section, une description plus au moins détaillée de ces deux algorithmes :

4.1.Les colonies d'abeilles

Les abeilles possèdent des propriétés assez différentes de celles des autres espèces d'insectes. Elles vivent en colonies, en construisant leurs nids dans des troncs d'arbres ou d'autres espaces clos similaires *[Yahya, 2007]*. Généralement, une colonie d'abeilles contient une femelle reproductrice appelée reine, quelques centaines de mâles connus sous le nom de faux-bourdon, et de 10.000 à 80.000 femelles stériles qui s'appellent les ouvrières. En se basant sur les comportements des abeilles, les scientifiques ont développé plusieurs algorithmes puissants. Nous représentons

Algorithme d'optimisation de colonie d'abeilles artificielle (ABC)

L'algorithme ABC (Artificiel Bee Colony) est développé par Karaboga et Basturk*[Karabogaand Basturkm, 2007]*, en inspectant les comportements des abeilles réelles pour trouver la source de nourriture, qui s'appelle le nectar, et partager l'information des sources de nourriture aux autres abeilles dans le nid.

Dans cet algorithme, les abeilles artificielle sont définies et classifiées en trois groupes : abeilles employeuses (abeilles qui recherche la nourriture), spectatrices (abeilles d'observation) et scouts (éclaireuses) sont chargées de trouver de nouvelles nourritures, (le nectar de nouvelles source).

Pour chaque source de nourriture, il y a seulement une abeille employeuse. C'est-à-dire, le nombre d'abeilles employeuses est égal au nombre de sources de nourriture *[Karabogaand Basturkm, 2007]*. Si l'abeille employeuse d'un site ne réussit pas de trouver la source de nourriture, elle doit être forcément devenir un scout pour rechercher aléatoirement de nouvelles sources de nourriture. Les abeilles employeuses partagent l'information avec les abeilles spectatrices dans une ruche de sorte que les abeilles spectatrices puissent choisir une source de nourriture pour l'explorer. Le processus de l'algorithme ABC est présenté comme suit:

1. **Etape 1- Initialisation :**On commence par sélectionner Fe pourcentage de population de façon aléatoire dans l'espace de recherche.

2. **Etape 2- Déplacement des abeilles employeuses:** Calculer la probabilité de choisir une source de nourriture, puis sélectionner une source de nourriture et ensuite déterminer ses quantités de nectar.
3. **Etape 3-Déplacer les scouts :** Si les valeurs de Fitness des abeilles employeuses ne sont pas améliorées par un nombre d'itérations prédéterminé, appelé ”max-cycle”, ces sources de nourriture sont abandonnées, et l'abeille trouvée dans cet emplacement passera aléatoirement pour explorer d’autres nouveaux emplacements. (Abeilles employeuses deviennent des Scouts).
4. **Étape 4-Mettre à jour la meilleure source de nourriture trouvée jusqu'ici :** Apprendre la meilleure valeur de Fitness et la position, qui sont trouvées par les abeilles, et les mémoriser.
5. **Étape 5-Critère d’arrêt Vérifier le processus de calcule jusqu’à** ce que le nombre d’itérations atteigne la valeur maximale prédéfinie ou qu’une solution de la fonction objective acceptable.

4.2.Marche Lévy walk

Lévy walk est une stratégie de recherche aléatoire, la trajectoire du processus de Lévy est caractérisée par un ensemble de petits pas reliés par un long pas. Selon *Tran et al.* [Tran et.al, 2004], les promenades de lévy sont des pas avec longueur aléatoire l réparties qui peut être approximativement caractérisé par la fonction de densité dans *l’équation 1*:

$$P(x) \sim |x| - 1 - \beta \, , avec \beta \; (0 < \beta \leq 2) \qquad (1)$$

On considère un processus de marche aléatoire à chaque pas de la longueur l avec la probabilité *l’équation 2*:

$$P(l) = \frac{\beta}{L0(1 + l/l0)1+\beta} \qquad (2)$$

La longueur des pas l peut être générée de manière aléatoire selon *l’équation 3*:

$$l = l_0 + \frac{1}{U^{1/\beta}}\text{-}1 \qquad (3)$$

OùU est un nombre flottant uniformément distribué dans un intervalle de $[0, 1]$.

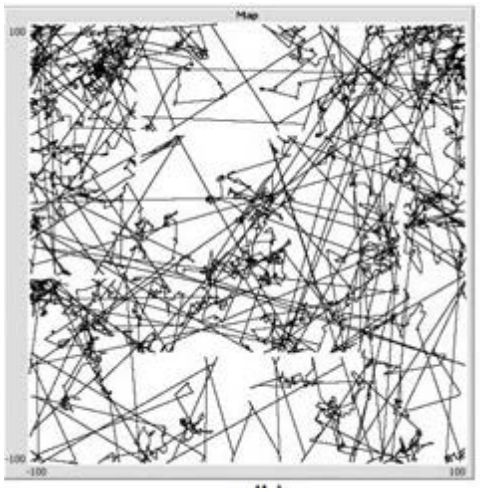


Figure2.1: Trajectoire de Lévy-Walk (avec $\beta = 2$) [Pina et.al, 2011]

Yang et Deb [Yang et Deb, 2010] montrent que l'utilisation d'une stratégie d'exploration basée sur Lévy-walk rend l'exploration globale de l'environnement plus efficace par rapport au modèle gaussien, en particulier dans les environnements inconnus.

5. Proposition d'un système adaptatif multi-dépôts bio-inspiré:

Nous proposons un nouveau modèle de foraging multi-dépôts adaptatif inspiré de comportement des abeillespour résoudre le problème de durabilité des dépôts (***AMB:*** ***A****adaptatif* ***M****ulti-nest* ***B****ee*).La distribution des objets en cluster, on utilisant plusieurs dépôts à partir desquels les robots partent et y retournent. On suppose que les robots ont des capteurs de proximité capable de détecter les obstacles et les autres robots et capteurs de lumière.

La méthode est inspirée du comportement des abeilles pour la recherche de nourriture et le déplacement des scouts est en Lévy-Walk. On utilise un recrutement simple par envoie de message entre robots. Nous prenons le principe général de comportement des abeilles et le principe général de l'algorithme ABC.

- *Les scouts :* cherchent les sources des objets (clusters) ;
- *Les employeurs:* récoltent les objets.

Nous avons remplacé les spectatrices de l'algorithme ABC par un vecteur de positions des sources.

5.1. Comportement des robots

1) Scouts

Les scouts commencent la recherche des sources des objets (clusters) par un mouvement en Levywalk avec $\beta=0,75$ (Cette valeur a été fixée expérimentalement dans [Idiri, 2018]) jusqu'à trouver les sources, cela permet d'effectuer de longs pas qui vont contribuer à attendre de nouvelles régions dans l'environnement.

Si un scout détecte un cluster, il récolte un objet et retourne vers le dépôt, il enregistre la position du cluster dans un vecteur au niveau de dépôt puis informe les robots employeurs qu'il a trouvé une source par un message de broadcaste (*Successful-Exploring*) et change son état àl'état *Reste*. Sinon si le scout continuedans l'état*chercher*pour une durée T_{max} sans trouver aucune source il revient vers le dépôt etchange son étatà*Reste* après l'envoi d'un message broadcasteaux employeurs leur informons qu'il n'a rien trouvé. Le scout reste à état *Reste* durant un temps T_{Reste},et puis il changeà l'état *scouter*.

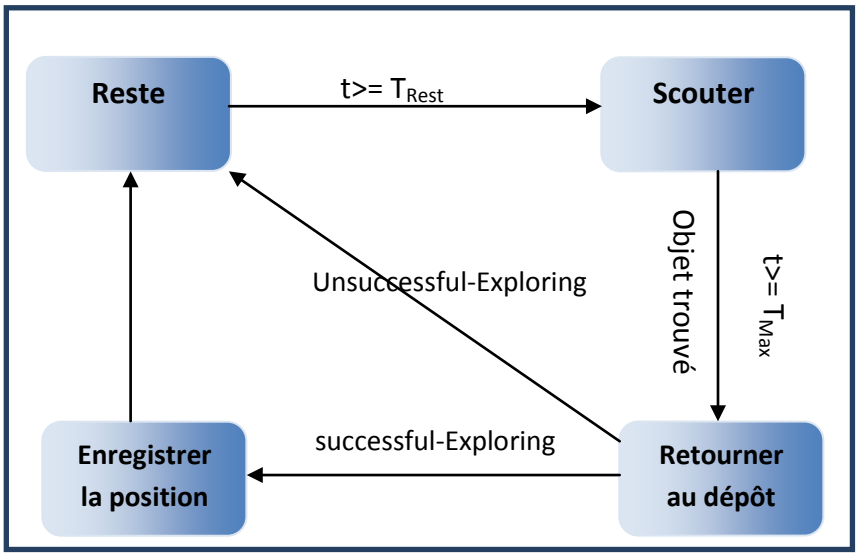


Figure 2.2 Machine d'état de comportement des scouts

2) Employeurs

Les robots employeurs passe de l'état *Reste* à l'état *Exploration*, lorsqu'ils reçoivent un message broadcaste "*Exploration_Succés*" envoyé par les scouts. Ils prennent alors une valeur probabiliste *P* dans l'intervalle de $[0, 0.5]$ et appliquent la règle d'alimentation utilisé dans l'exemple de base de foraging d'ARGoS, en augmentant la probabilité

"*RestToExplor*"(*équation 4*) et diminuant la probabilité "*ExplorToRest*" (*équation 5*) pour diminuer le taux de collision au niveau de dépôt et au niveau du cluster.

$$P_{RestToExplor} = P_{RestToExplor} + P_{Delta} \; avec \; P_{Delta} = 0.01 \qquad (4)$$

$$P_{ExplorToRest} = P_{ExplorToRest} - P_{Delta} \; avec \; P_{Delta} = 0.01 \qquad (5)$$

Les robots employeurs calculent la position du cluster le plus proche au dépôt parmi les positions de clusters dans le vecteur. Puis ils se déplacent tout droite vers le cluster sélectionné, lorsqu'ilsarrivent à la position du cluster ils appliquent une recherche local en Lévy-Walk avec $\beta=2.25$ (cette valeur a été fixée expérimentalement, voir chapitre 3) durant un certain temps T_{min}.

Le robot passe à l'état *retournerau dépôt* dans deux situations :

1. Si le robot à une cible. Il applique la règle d'alimentation précédente et retourne audépôtaprès changer la valeur de dernière exploration à "*Dernière_Exploration_Succés*".
2. Si le robot n'a pas trouvé une cible depuis un certain temps, le changement de l'état est probabiliste. Le robot prend une valeur probabiliste aléatoire$P$ dans un intervallede $[0, 1]$.
   - Si $P < P_{ExplorToRest}$ le robot retourne vers le dépôt et change la valeur de la dernière exploration *à "Dernière_Exploration_Manqué "*.
   - Sinon, le robot applique la règle d'alimentation, en diminuant la probabilité "*RestToExplor*" et augmentant la probabilité "*ExplorToRest*" jusqu'à $P < P_{ExplorToRest}$ devient supérieure à P.

Lorsque le robot arrive au dépôt avec un objet, il passe à l'état *Reste* après l'envoid'un message broadcaste de l'état de la dernière exploration "*Dernière_Exploration_Succés*".

Dans le cas où le robot employeur n'a rien trouvé, il détruit la position du cluster dans le vecteur des positions des sources au niveau de dépôt puis passe à l'état "*Reste*" après l'envoid'un message broadcaste de l'état de la dernière exploration*"Dernière_Exploration_Manqué"*.Le robot reste en état "*Reste*" pendant une période de temps T_{Reste}, il augmente la valeur $P_{RestToExplor}$ et calcule la position des clusters enregistrés dans le vecteur au niveau du dépôt pour choisir le cluster le plus proche par rapport au dépôt et passe à l'état "*exploration*".

Dans le cas où le vecteur des positions des clusters est vide les robots restentàl'état "*Reste*" jusqu'à recevoir un message broadcaste par l'un des scouts.

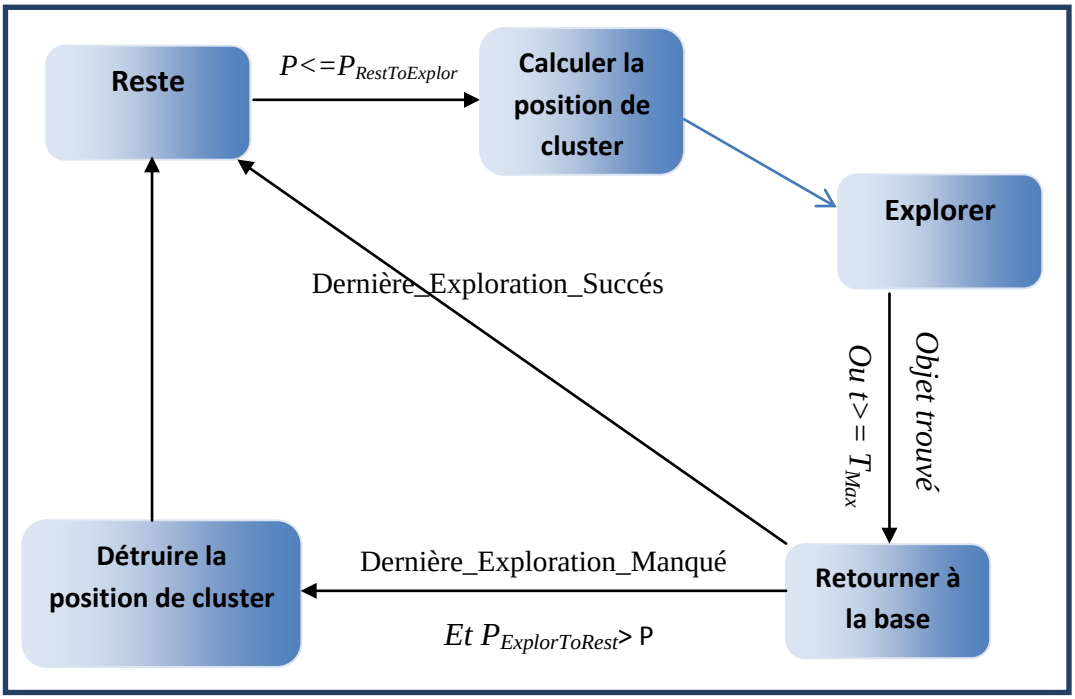


Figure 2.3: Machine d'état de comportement des employeurs

3) Gestion de la durabilité de dépôt

Lorsque la quantité des objets dans le dépôt arrive à une quantité minimale, le dépôt change son état à l'état"alerte". Lorsque le robot passe de l'état "*Reste*" à l'état "*Explore*", il allume ses LED en couleur orange et devient "*Recruteur*", il envoie ainsi dans sa période d'exploration un message broadcaste contenant la position de son dépôt.

Chaque robot a une caméra qui capte les couleurs des LED des autres robotsdepuis une distance d. Si le robot détecte un robot avec couleur orange, il le recrute et devient "*Recruteur*" après qu'il reçoit la position de dépôt en alerte. Le robot recruté continue à collecter des objets pour le dépôten alertejusqu'à le stock dans le dépôt dépasse la quantité minimale.

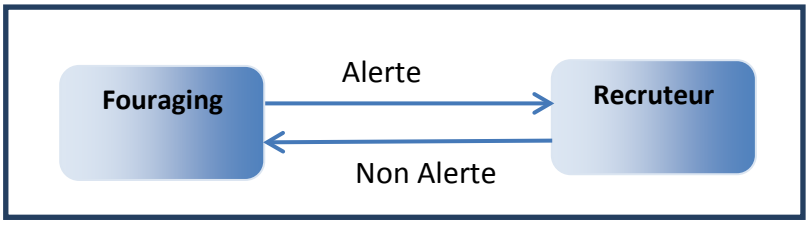


Figure 2.4: Automate d'état des robots pour gérer la durabilité des dépôts

5.2. Organigramme de l'algorithme

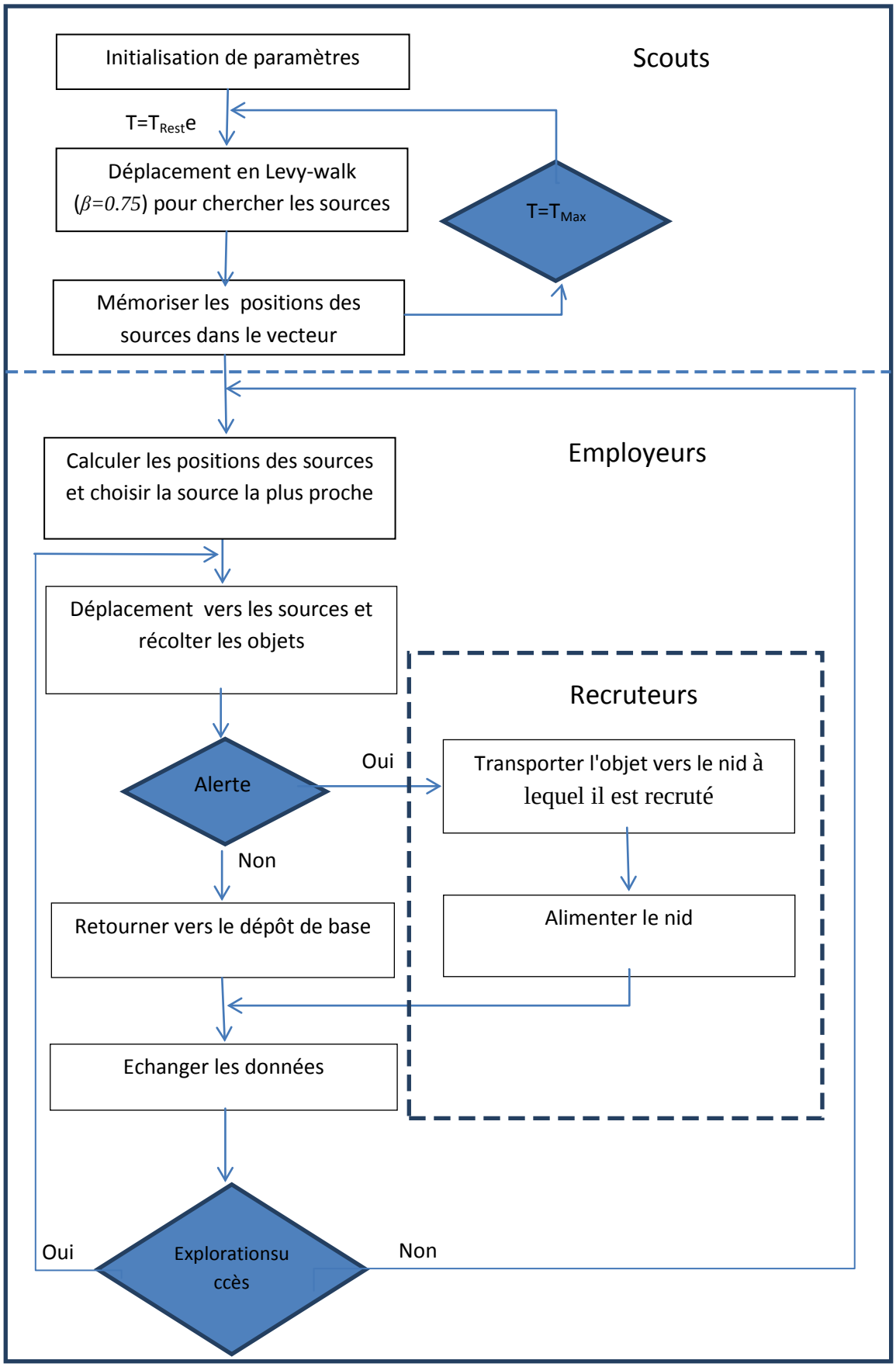


Figure 2.4:Architecture général de "AMB"

5.3. Pseudo-code de l'algorithme "*AMB*"

Générer la population initiale de taille n;

Scouts:
Déplacer les scouts;
Appliquer la stratégie d'exploration globale en Lévy-Walk avecβ=0.75;
Retourner au dépôt si trouver une source ou si t= T_{Reste};
Mémoriser les positions des sources dans le vecteur ;
Aller à scouts, après t = T_{Reste}

Employeurs :
Calculer les positions des sources et choisir les sources les plus proche;
Déplacement des robots employeurs vers meilleure source;
Faire une exploration locale en Levy-walk avec β=2.25;
Si alerte alors
Recruter les robots rencontrés dans le trajet;
fin si
Si objet trouvé alors
Récolter l'objet;
Si alerte alors
Transporter la cible vers le nid en alerte;
Sinon
Transporter l'objet a la base
fin si
Sinon
Retour vers le nid auquel le robot appartient;
fin si
Vérifier l'état de dépôt;
Si alerte alors état-robot = recruteur;
Sinon état-robot=foraging
fin si

Si Dernière-Exploration =" Exploration_Succés" alors
Aller àEmployeur;
Sinon
Aller à l'état "Reste";
fin si
FIN

6. Conclusion:

L’intelligence en essaim est maintenant reconnue comme un paradigme robuste et flexible pour faire face aux problèmes distribués dans des environnements complexes. Nous nous intéressons aux groupes de robots avec des capacités de calcul, de perception et de mémorisation limitées.

Nous avons présenté dans ce chapitre une approcheadaptative bio-inspirébasée sur le comportement des abeilles pour résoudre le problème de durabilité desdépôts (satisfaire les demandes d'un système externe). Les robots recherchent des objets pour leur dépôts d’origine selon une stratégie d’exploration stratégique. Pour assurer et maintenir la durabilité (survie) du système, les robots peuvent être recrutés à d’autres dépôts en état d’alerte.

En vue de valider l’efficacité de l'approche proposée, nous allons dans le chapitre suivant, montré l’implémentation et les testspuis une comparaison quantitative de ce qui a été proposé dans ce chapitre. La description des outils développés, les simulations réalisées, les résultats obtenus ainsi les comparaisons feront l’objet du chapitre suivant.

CHAPITRE 3

CHAPITRE 3

Implémentation et Expérimentations

1. Introduction

Ce chapitre a été consacré à l'implémentation, l'étude expérimentale, et les discussions des résultats obtenues par l'algorithme proposé dans le chapitre précédent. Notre proposition a été implémentée sous la plateforme de robotique mobile ARGoS. Nous avons réalisé plusieurs simulations dans des configurations environnementales différentes.

Nous commençons ce chapitre, par présenter la plateforme ARGoS. Par la suite nous décrivons les caractéristiques des composantes de notre système de foraging. Nous proposons aussi quelques scénarios de simulation. Puis, nous présentons et discutons les résultats obtenus par l'algorithme proposé. Nous terminerons le chapitre par une conclusion.

2. Environnement de développement

ARGoS (Autonomous Robots Go Swarming)[Pinciroli, 2012], est un simulateur de robots crée par Carlo Pinciroli pendant le projet Swarmanoid. Il est développé dans le laboratoire IRIDIA. Le but de ce projet était de développer un outil flexible et efficace afin de simuler des expériences complexes impliquant des essaims de robots de différents types. ARGoS s'inscrit dans une optique aussi flexible qu'efficace. Il se veut en effet être flexible grâce à la modularité introduite dans celui-ci : ses modules, vus comme des plugins peuvent être configurés, étendus, inters changés suivant les nécessitées des différentes expériences. Différentes implémentations des mêmes composants peuvent même exister afin d'obtenir différents niveaux de précisions, et leurs coûts correspondants. D'un autre côté, l'efficacité se retrouve dans le fait de ne calculer que les composants nécessaires, tout en créant une architecture parallèle afin de pouvoir faire usage des processeurs multi-cœurs. ARGoS est un simulateur disponible de manière libre et gratuite et est utilisé par de plus en plus de laboratoires à travers le monde.

Comme dans les programmes les plus complexes, ARGoS découpe son architecture en de nombreux modules interagissant. C'est en effet le moyen le plus répandu afin de permettre à l'utilisateur de choisir les modules à utiliser, et éventuellement de modifier ceux existants, ou d'en ajouter de nouveau.

L'avantage de cette modularisation est la possibilité de permettre l'activation d'uniquement les modules qui sont utiles pour la simulation, et ainsi de ne dépenser que les ressources nécessaires à la simulation. Cette modularité peut s'observer sur la figure 3.1où l'on peut observer que le robot (entité), les capteurs et actionneurs, le Controller (logique du robot) sont des modules configurables par l'utilisateur, mais également le moteur physique, la visualisation et la 'loopfunction'(logique supplémentaire pouvant accéder à tous les éléments de la simulation).

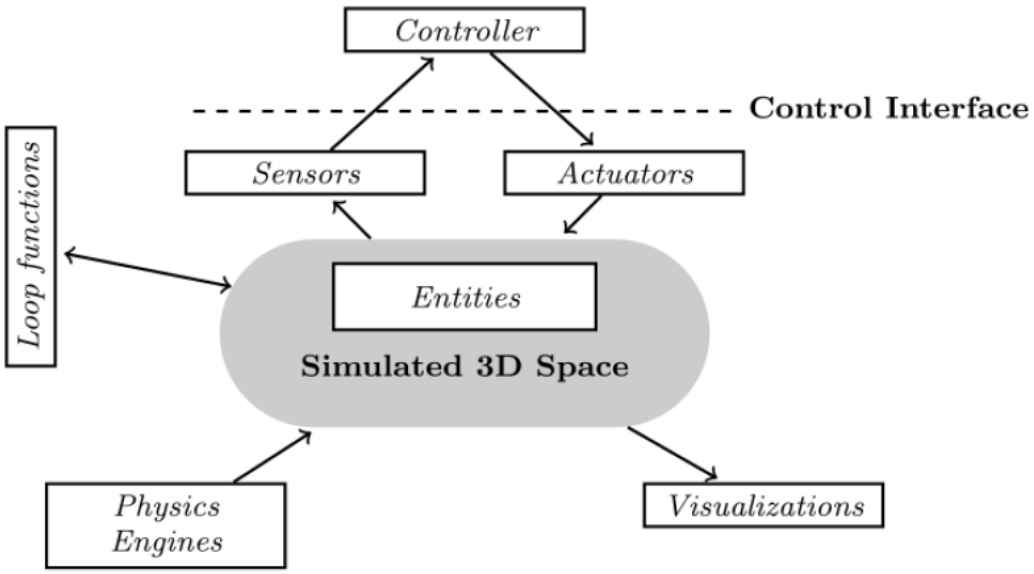


Figure 3.1. Architecture d'ARGoS. Chaque boite blanche correspondant

3. Modélisation des composants de notre système

Notre système Multi-Robots se compose de quatre composants principaux: l'environnement, les robots, les objets et les dépôts. Nous montrons dans ce que suit la modélisation proposée pour chacun de ces composants :

1. L'environnement est modélisé par un espace 3D.
2. Les robots sont des Foot-bots (Voir *section 3.1*).
3. Les objets sont représentés par des cercles noirs.
4. Les dépôts sont représentés par des carrés gris.

3.1. Les Foot-bots

Le foot-bot est une configuration particulière de modules basée sur la plate-forme robotique marXbot. La configuration de foot-bot comprend un processeur supérieur et un module de

vision, un scanneur de distance, un module de course et de relèvement, un module à assembler par soi-même et un module de base (mobilité et batterie).

Caractéristiques de Foot-bot:

- Taille compacte: 17 cm de diamètre x 29 cm, 1,8 kg
- Système modulaire
- Puissance de calcul élevée, y compris le traitement flottant
- Scanner de distance mesurant une distance allant jusqu'à 1,5 m
- Deux caméras, l'une montée en omnicamera
- Mécanisme d'auto-assemblage
- Batterie remplaçable à chaud
- Système de mobilité Treel

Les capteurs:

Nous présentons les déférents capteurs existent chez le Foot-bot

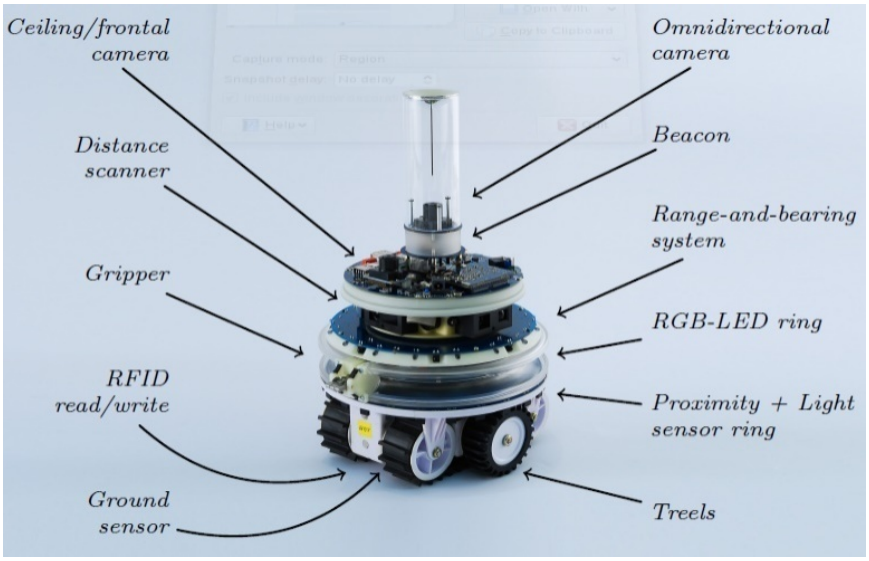
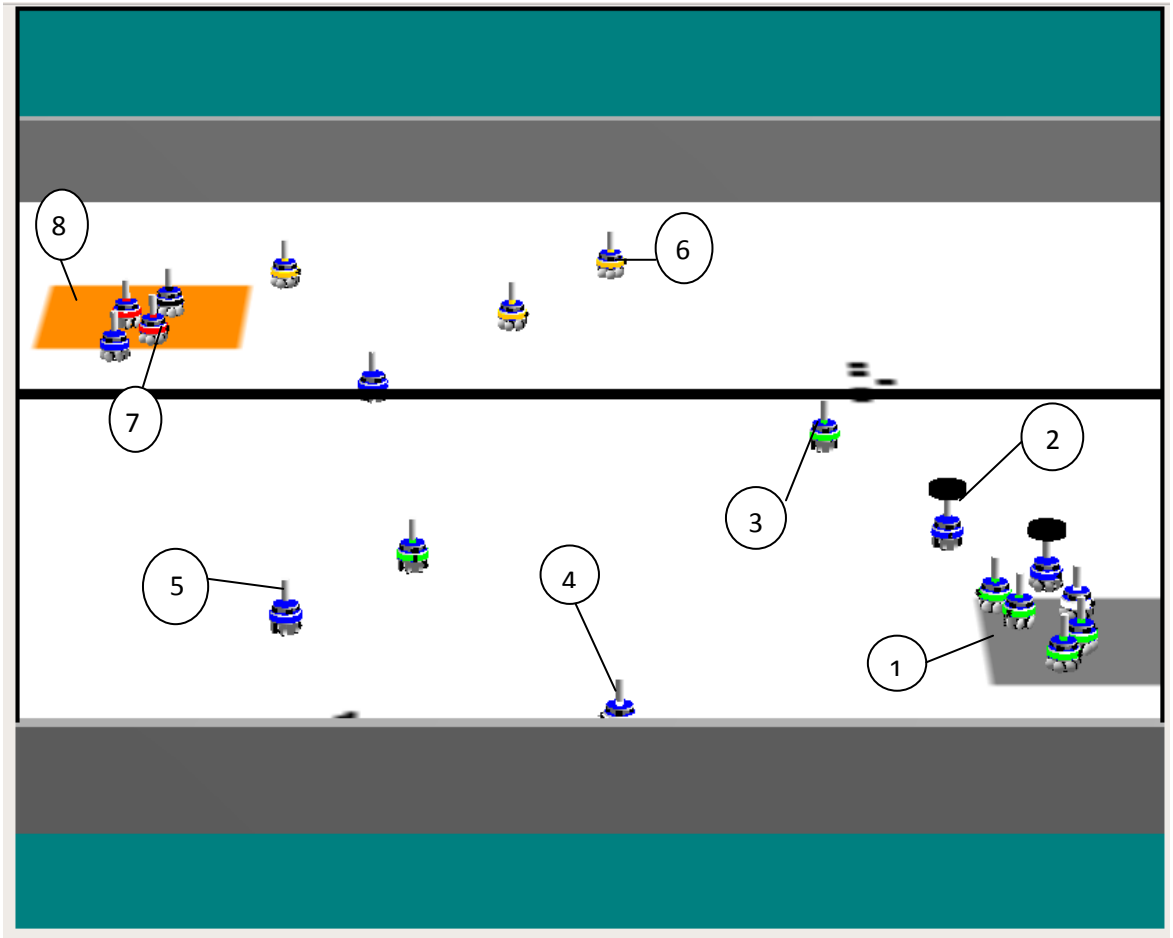


Figure 3.2. Capteurs de robot Foot-bot

Le Foot-bot se déplace dans l'environnement en utilisant une paire de roues. Dans le code de contrôle, on utilise la procédure " *set_velocity (l, r)*" pour le déplacement des robots, où "l" et "r" sont respectivement les vitesses de roue gauche et droite. Pour détecter les obstacles et collisions nous utilisons les 24 capteurs de proximité où chaque capteur a une portée de $0.1m$.La caméra`"colored_blob_omnidirectional_camera"` permet au robot de détecter les couleurs de LED des autres robots, si le robot détecte une couleur orange du LED, il change sa couleur de LED en orange et il devient "recruteur". Pour trouver les objets qu'on a représentés par des cercles noirs nous utilisons les 4 capteurs "*motor_ground* ".

4. Interface du système

La *Figure 3.3*montre une exécution de l'algorithme de foraging proposé sous ARGoS.



1. Dépôt en état stable
2. Foot-bot transporte un objet
3. Employeur se déplace
4. Scout
5. Employeur en état "Dernière_Exploration_Manqué"
6. Foot-bot recruteur
7. Employeur en état "Reste"
8. Dépôt en état "Alerte"

Figure 3.3. Une exécution de l'algorithme*AMB* implémenté sur ARGoS

5. Simulation et analyse des résultats

5.1. Etude l'influence de la valeur de *"Beta"* de Levy-walk pour la recherche locale:

Pour assurer une bonne qualité des résultats, nous avons réalisé une étude pour choisir la valeur de "*Beta*" dans l'équation utilisée pour calculer la longueur des pas dans la phase de recherche (équation (3), voir chapitre 2). Le scénario utilisé pour choisir cette valeur est le suivant :

- Taille de l'environnement $10\ m \times\ 10\ m$
- Nombre des dépôts = 2;
- Nombre des robots employeurs pour chaque dépôt: 4;
- Nombre des scouts pour chaque dépôt: 2;
- Nombre de clusters: 6;
- Nombre des objets: 216;
- Vitesse de déplacement : 20 cm/s;
- Valeurs de beta: 1.75, 2, 2.25, 2.5;

Les résultats dans le*Tableau3.1* montrent le nombre d'objets trouvés dans un temps fixe, pour différentes valeurs de $beta$.

Minutes / Beta	5	10	15	20
1,75	*37*	*72*	*104*	*134*
2	*36*	*75*	*113*	*147*
2,25	*41*	*79*	*123*	*152*
2,5	*37*	*77*	*115*	*152*

Tableau 3.1. Influence de la valeur de Beta sur les performances

Depuis les résultats montrés dans le *Tableau 3.1*, on peut déduire que le nombre maximum des objets dans les dépôts est donnée avec$beta\ =\ 2.25$. Donc dans le reste des simulations nous allons fixer la valeur de$beta$ à 2.25.

5.2. Critères de performance

Nous avons utilisé les deux critères de performances suivants:

- Satisfaction des demandes externes, donc il faut que la quantité des objets dans les dépôts toujours supérieure ou égale les demandes.
- Le nombre des objets trouvés dans un temps fixe(*15 min*).

5.3. Scenarios de simulation

Nous avons réalisé six scénarios de simulation, dans toutes les simulations nous avons proposé que la quantité initiale de chaque dépôt soit sept objets.Les six scénarios réalisés sont résumés dans le *Tableau 3.2*.

Scénario1: Variation des unités de consommation	Scénario2 : Variation de nombres d'objets
Taille de l'environnement $10\ m\ x\ 10\ m$ Nombre des dépôts: 2; Nombre des robots employeurs par dépôt: 4; Nombre des scouts pour chaque dépôt: 2; Nombre de clusters: 6; Nombre des objets: 216; Vitesse: 20 cm/s; Taux de consommation: 1d/15s, 1d/20s	Taille de l'environnement $10\ m\ x\ 10\ m$ Nombre des dépôts: 2; Nombre des robots employeurs par dépôt: 4; Nombre des scouts pour chaque dépôt: 2; Nombre de clusters: 6, 7, 8; Nombre des objets: 216, 252, 288; Vitesse: 20 cm/s; Taux de demandes: 1d/15s,
Scénario3: Variation de nombre des dépôts	Sénario4: Variation de la vitesse des robots
Taille de l'environnement $10\ m\ x\ 10\ m$ Nombre des dépôts: 2, 3, 4; Nombre des robots employeurs par dépôt: 4; Nombre des scouts pour chaque dépôt: 2; Nombre de clusters: 6; Nombre des objets: 216; Vitesse: 20 cm/s; Taux de demandes: 1d/20s;	Taille de l'environnement $10\ m\ x\ 10\ m$ Nombre des dépôts: 4; Nombre des robots employeurs par dépôt: 4; Nombre des scouts pour chaque dépôt: 2; Nombre de clusters: 6; Nombre des objets: 216; Vitesse: 20 cm/s, 25 cm/s, 30cm/s; Taux de demandes: 1d/20s
Scénario5: Variation de la taille de l'environnement	Scénario6: Variation du nombre des robots
Taille de l'environnement $10 \times 10\ m,\ 15 \times 15m$ Nombre des dépôts: 4; Nombre des robots employeurs par dépôt: 4; Nombre des scouts pour chaque dépôt: 2; Nombre de clusters: 8; Nombre des objets: 288; Vitesse: 20 cm/s, 30 cm/s; Taux de demandes: 1d/20s	Taille de l'environnement $15 \times 15m$ Nombre des dépôts: 4; Nombre des robots employeurs par dépôt: 4,6,8; Nombre des scouts pour chaque dépôt: 2,3,4; Nombre de clusters: 8; Nombre des objets: 288; Vitesse: 30 cm/s; Taux de demandes: 1d/20s

Tableau 3.2. Scénarios de simulations réalisés

1) Scénario 1

Ce scénario traite l'influence de taux de demandes par un système externe sur la performance de l'algorithme *AMB* pour satisfaire ces demandes. Dans un premier temps nous avons supposé qu'il y ait une demande chaque 20 secondes par dépôt (3 demandes chaque minute

pour chaque dépôt, puis nous avons faitun test avec une demande chaque 15 secondes pour chaque dépôt (4 demandes chaque minute pour chaque dépôt).

Minutes / Dépôts	2,5	5	7,5	10	12,5	15
Dépôt 1	*5*	*5*	*8*	*13*	*17*	*18*
Dépôt 2	*10*	*14*	*17*	*18*	*20*	*18*

Tableau 3.3. Influence de 3 consommations/minute

Minutes / Dépôts	2,5	5	7,5	10	12,5	15
Dépôt 1	*2*	*4*	*7*	*1*	*2*	*4*
Dépôt 2	*7*	*4*	*5*	*4*	*8*	*9*

Tableau 3.4. Influence de 4 consommations/minute

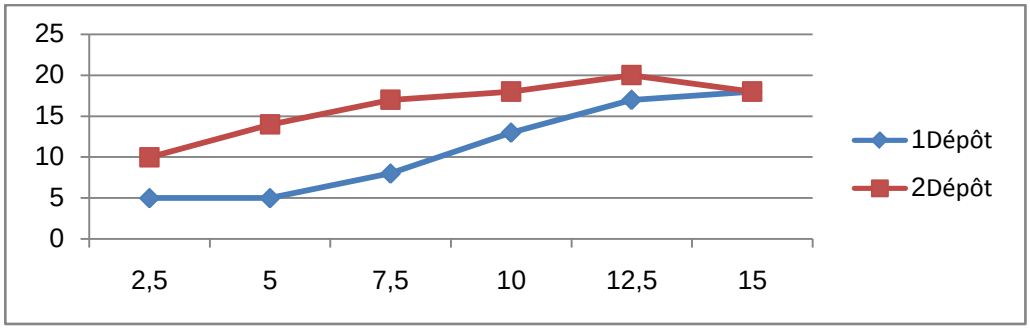


Figure 3.4. Influence de 3 consommations/minute

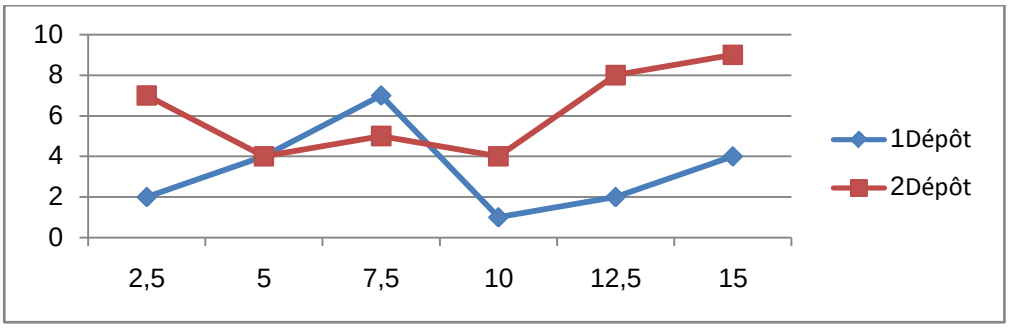


Figure 3.5. Influence de 4 consommations/minute

Discussion:

Les résultats dans le Tableau 3.3 et Tableau 3.4 et les Figures 3.4 et 3.5montrent que la performance de l'algorithme diminue en fonction de taux de demandes. Avec 3 demande/minutes, *AMB* a satisfait les demandes durant toute la période de test et n'a jamais appliqué la politique de recrutement des robots. Avec 4 demandes/minute, nous avons remarqué des perturbations après la dixième minute, même dans quelque tests la quantité des

objets tombe à zéro durant quelques secondes, mais aucune demande n'a été ratée. Le système a récupéré sa stabilité grâce à la politique de gestion de la durabilité des dépôts.

2) Scénario 2:

Ce scénario montre l'influence du nombre des objets et des clusters sur la performance de l'algorithme AMB dans un environnement avec 2 dépôts. Nous testons avec 216 objets (6 clusters), 252 objets (7 clusters) et 288 objets (8 clusters).

Clusters	Minutes / Dépôts	2,5	5	7,5	10	12,5	15
6 clusters	Dépôt 1	*2*	*4*	*7*	*1*	*2*	*4*
	Dépôt 2	*7*	*4*	*5*	*4*	*8*	*9*
7 clusters	Dépôt 1	*3*	*4*	*5*	*7*	*7*	*12*
	Dépôt 2	*7*	*3*	*12*	*11*	*16*	*12*
8 clusters	Dépôt 1	*2*	*7*	*8*	*9*	*11*	*9*
	Dépôt 2	*8*	*8*	*13*	*13*	*17*	*14*

Tableau 3.5:Influence de nombre d'objets dans l'environnement

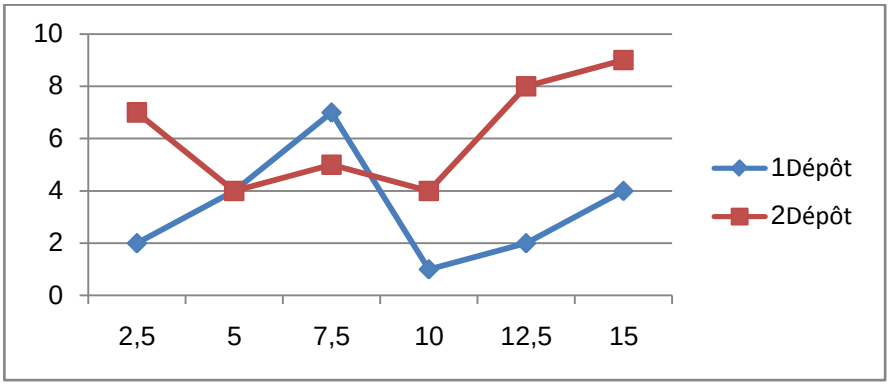


Figure 3.6. Influence de nombre d'objets dans l'environnement avec 216 objets

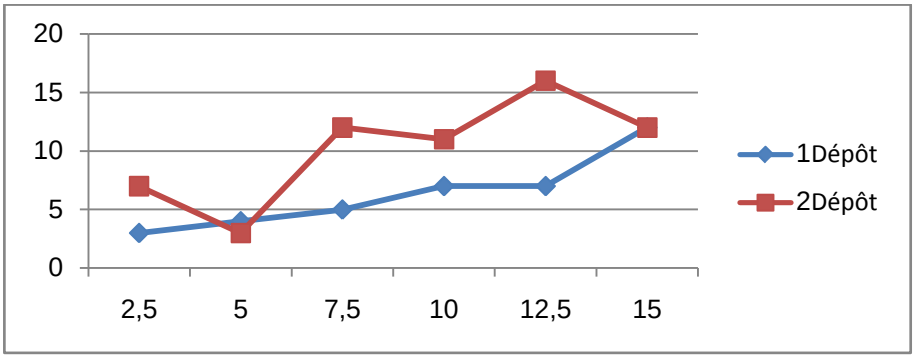


Figure 3.7. Influence de nombre d'objets dans l'environnement avec 252 objets

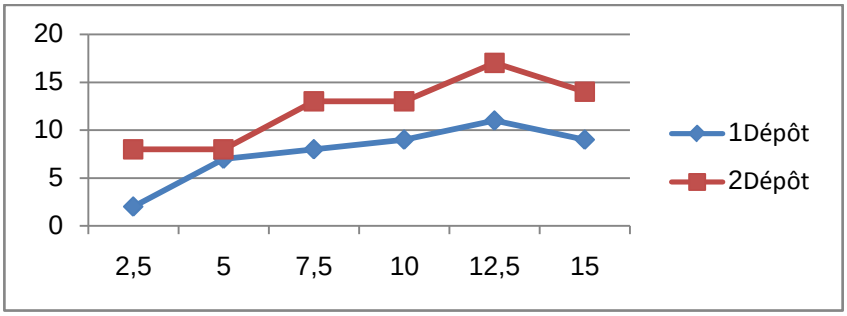


Figure 3.8. Influence de nombre d'objets dans l'environnement avec 288 objets

Discussions:

LeTableau 3.5, et les Figures 3.6, 3.7, 3.8 montrent les résultats obtenus avec l'algorithme *AMB* lorsqu'on varie le nombre des objets dans l'environnement. Avec plus d'objets dans l'environnement le système de foraging est plus stable et assure une quantité suffisante des objets dans les dépôts .Nous remarquons dans les 3 scénarios qu'il y a des alertes au niveau des dépôts dans les trois premières minutes, parce que les robots employeurs sont en état "*Reste*" jusqu'à la localisation des sources par les scouts, Le système a récupéré sa stabilité avec la stratégie de location des robots dans le cas où l'environnement contient 216 et 252 objets. Avec 288 objets dans l'environnement le système a récupéré sa stabilité sans recrutement par ce que les scouts ont trouvé rapidement des nouvelles ressources.

3) Scénario 3

Ce scénario test la performance de système de foraging proposé quand on augmente le nombre des dépôts et des robots, chaque dépôt a 4 robots employeurs et 2 robots scouts, le taux de demandes est une demande chaque 20 seconde pour chaque dépôt, donc le système répond à 9 demandes chaque minute pour simulation de 3 dépôts et 12 demandes chaque minute pour 4 dépôts. Nous comparons les résultats avec ceux obtenus en deux dépôts quand le taux de demandes est 8 demandes chaque minute.

Nbr dépôts	Minutes / Dépôts	2,5	5	7,5	10	12,5	15
Environnement avec 2 dépôts	Dépôt 1	*2*	*4*	*7*	*1*	*2*	*4*
	Dépôt 2	*7*	*4*	*5*	*4*	*8*	*9*
Environnement avec 3 dépôts	Dépôt 1	*1*	*9*	*15*	*22*	*25*	*25*
	Dépôt 2	*1*	*5*	*14*	*10*	*10*	*6*
	Dépôt 3	*7*	*12*	*11*	*13*	*11*	*10*
Environnement avec 4 dépôts	Dépôt 1	*13*	*17*	*25*	*26*	*21*	*16*
	Dépôt 2	*1*	*5*	*13*	*20*	*16*	*15*
	Dépôt 3	*13*	*18*	*17*	*21*	*17*	*10*
	Dépôt 4	*1*	*4*	*7*	*9*	*10*	*11*

Tableau 3.6. Influence de nombre des dépôts sur les performances

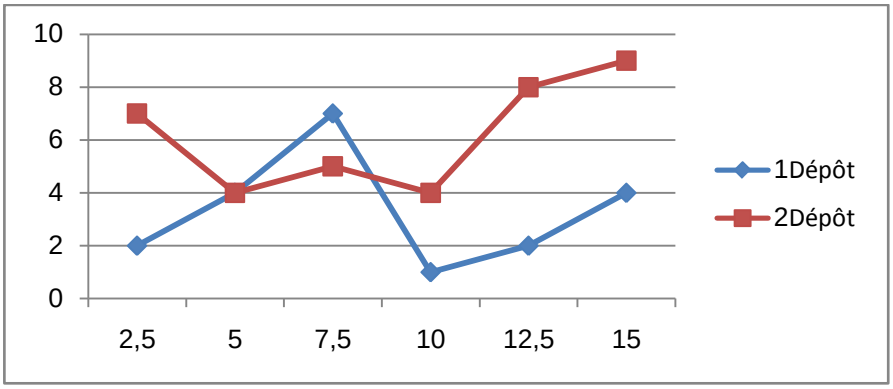


Figure 3.9. Influence de nombre des dépôts sur les performances (2 dépôts)

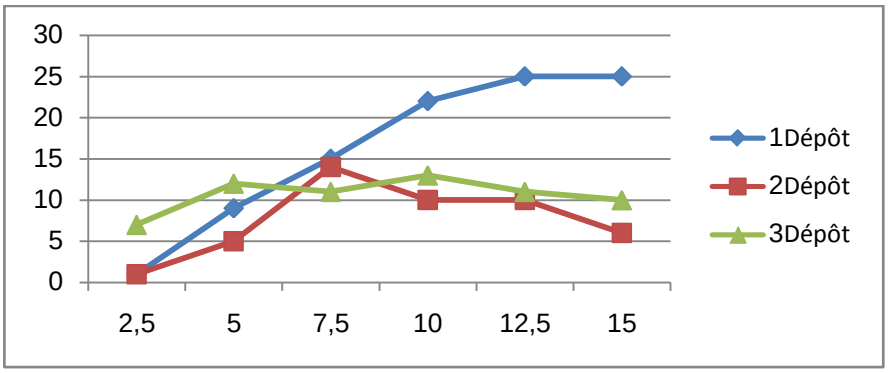


Figure 3.10. Influence de nombre des dépôts sur les performances (3 dépôts)

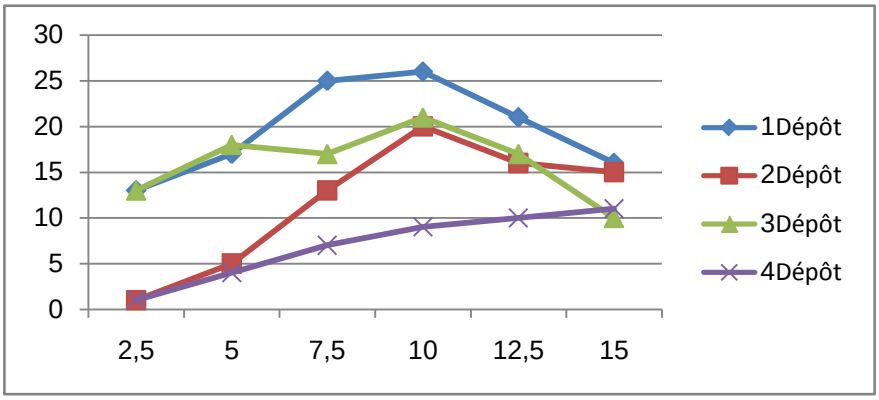


Figure 3.11. Influence de nombre des dépôts sur les performances (4 dépôts)

Discussion:

Le Tableau 3.6 et les Figures 3.9, 3.10, 3.11 montrent les résultats obtenus dans le troisième scénario. Pour satisfaire l'augmentation des demandes externes, il est préférable d'augmenter le nombre des dépôts pour équilibrer la charge des demandes. Nous remarquons dans tous les scénarios qu'il y a des perturbations dans quelque dépôt dans les trois premières minutes, parce que les robots employeurs sont en état "*Reste*" jusqu'à la localisation des sources par

les scouts, donc il est préférable que le système de foraging proposé n'accepte pas les demandes jusqu'à ce que tous les robots commencent la récolte des objets.

4) Scénario 4

Ce scénario traite l'influence de la vitesse des robots sur la performance de l'algorithme AMB. Nous augmentons la vitesse des robots de 20cm/s à 25cm/s puis à 30cm/s.Nous avons testé ce scénario durant des périodes de temps allons de 2,5 minutes jusqu'au 10 minutes.

Vitesse	Minutes / Dépôts	2,5	5	7,5	10
V= 20 cm/s	Dépôt 1	*13*	*17*	*25*	*26*
	Dépôt 2	*1*	*5*	*13*	*20*
	Dépôt 3	*13*	*18*	*17*	*21*
	Dépôt 4	*1*	*4*	*7*	*9*
	Objetscollectés	*28*	*76*	*122*	*168*
V= 25 cm/s	Dépôt 1	*17*	*17*	*20*	*15*
	Dépôt 2	*6*	*10*	*14*	*13*
	Dépôt 3	*7*	*14*	*26*	*26*
	Dépôt 4	*13*	*20*	*19*	*26*
	Objetscollectés	*43*	*93*	*139*	*169*
V= 30 cm/s	Dépôt 1	*16*	*12*	*12*	*13*
	Dépôt 2	*14*	*16*	*20*	*17*
	Dépôt 3	*7*	*18*	*26*	*29*
	Dépôt 4	*17*	*23*	*33*	*33*
	Objetscollectés	*53*	*101*	*151*	*184*

Tableau 3.7.Influence de la vitesse des robots sur les performances

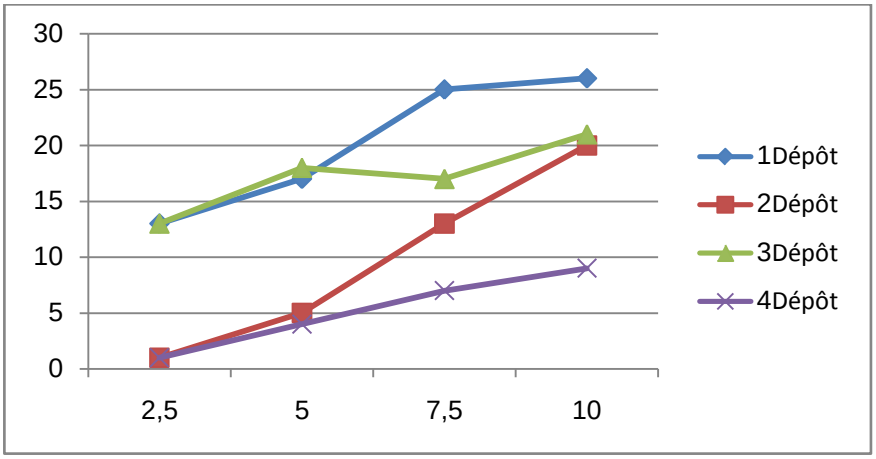


Figure 3.12. Influence de la vitesse des robots sur les performances (V= 20cm/s)

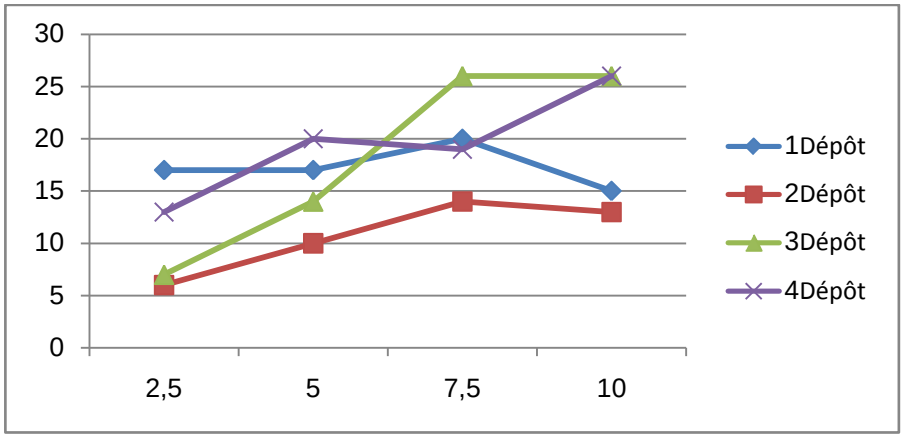


Figure 3.13. Influence de la vitesse des robots sur les performances (V= 25cm/s)

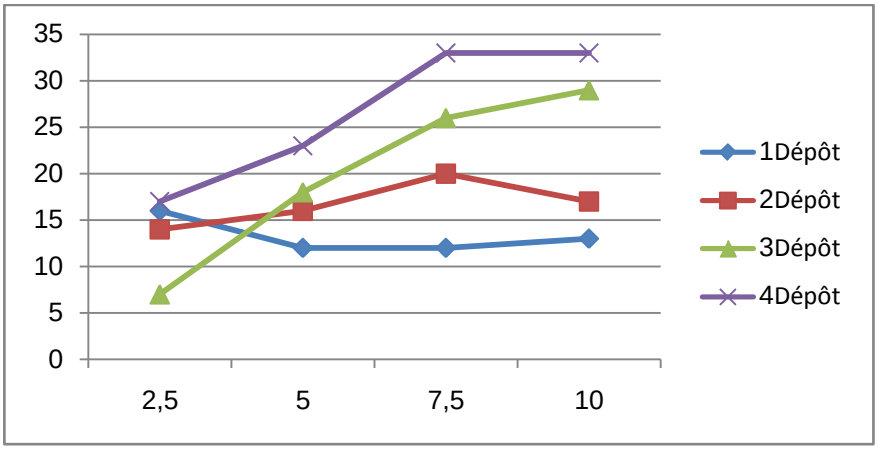


Figure 3.14. Influence de la vitesse des robots sur les performances (V= 30cm/s)

Discussion:

Le Tableau 3.7 et les Figures 3.12, 3.13, 3.14 montrent les résultats obtenus dans le scénario 4. Les résultats obtenus avec l'algorithme *AMB* montrent que le nombre des objets collectés augmente avec toute augmentation de la vitesse des robots, ce qui permet d'assurer une quantité suffisante des objets au niveau des dépôts. La Figure 3.12 montre une alerte de dépôt 2 jusqu'a cinquième minute par ce que les scouts de se dépôt ont pris plus de temps pour trouver les ressources, grâce a la politique de recrutement des robots, la quantité des objets est augmenté rapidement dans le dépôt 2 après la cinquième minute.

5) Scénario 5

Ce scénario traite l'influence de la taille de l'environnement sur la performance de l'algorithme *AMB*. Nous augmentons la taille de l'environnement de $10 \times 10m$ à $15 \times 15m$ et nous comparons les résultats obtenus dans le Tableau 3.13 avec les résultats de Tableau 3.12.

Taille de l'environnement	Minutes / Dépôts	2,5	5	7,5	10
10 x 10	Dépôt 1	*16*	*12*	*12*	*13*
	Dépôt 2	*14*	*16*	*20*	*17*
	Dépôt 3	*7*	*18*	*26*	*29*
	Dépôt 4	*17*	*23*	*33*	*33*
15 x 15	Dépôt 1	*9*	*29*	*33*	*30*
	Dépôt 2	*7*	*14*	*16*	*13*
	Dépôt 3	*9*	*14*	*16*	*16*
	Dépôt 4	*14*	*17*	*17*	*14*

Tableau 3.8.Influence de la taille de l'environnement sur les performances

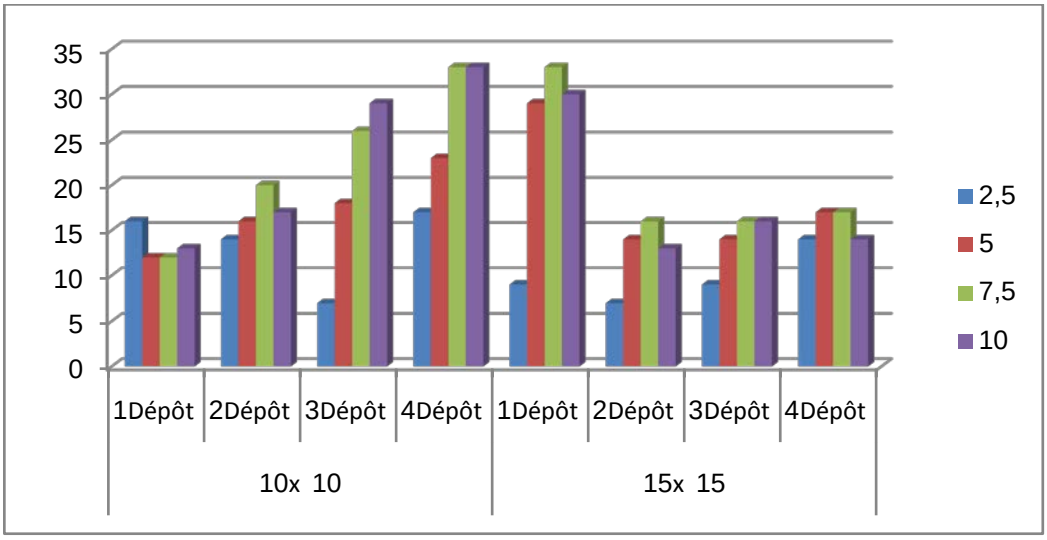


Figure 3.15. Influence de la taille de l'environnement sur les performances

Discussion:

Selon les résultats obtenus dans le Tableau 3.8 et la Figure 3.15, le nombre d'objets collectés diminue en fonction de l'augmentation de la taille de l'environnement, mais notre système de foraging a satisfait toutes les demandes avec une vitesse des robots suffisante sans aucune alerte par les dépôts.

Scénario 6:

Nous avons testé dans ce scénario l'influence de nombre des robots sur les performances de l'algorithme *AMB*. On augmente à chaque fois le nombre des robots employeurs et le nombre des robots scouts, où le nombre des employeurs est le double du nombre de scouts.

Nbr des robots	Minutes / Dépôts	2,5	5	7,5	10	12,5	15
4 employeurs et 2 scouts	Dépôt 1	9	29	33	30	30	28
	Dépôt 2	7	14	16	13	11	6
	Dépôt 3	9	14	16	16	21	18
	Dépôt 4	14	17	17	14	7	4
6 employeurs et 3 scouts	Dépôt 1	17	25	29	35	36	37
	Dépôt 2	1	3	1	4	3	2
	Dépôt 3	20	22	16	11	5	1
	Dépôt 4	14	24	22	19	14	11
8 employeurs et 4 scouts	Dépôt 1	16	28	43	46	41	39
	Dépôt 2	23	28	47	43	38	31
	Dépôt 3	25	30	39	42	47	44
	Dépôt 4	11	12	18	19	13	6

Tableau 3.9. Nombre total des objets dans les dépôts

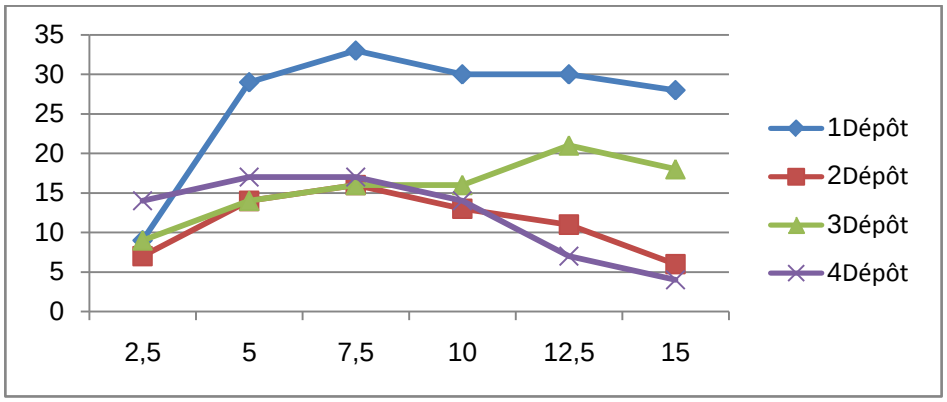


Figure 3.16. Influence du nombre des robots sur les performances (4 robots et 2 scouts)

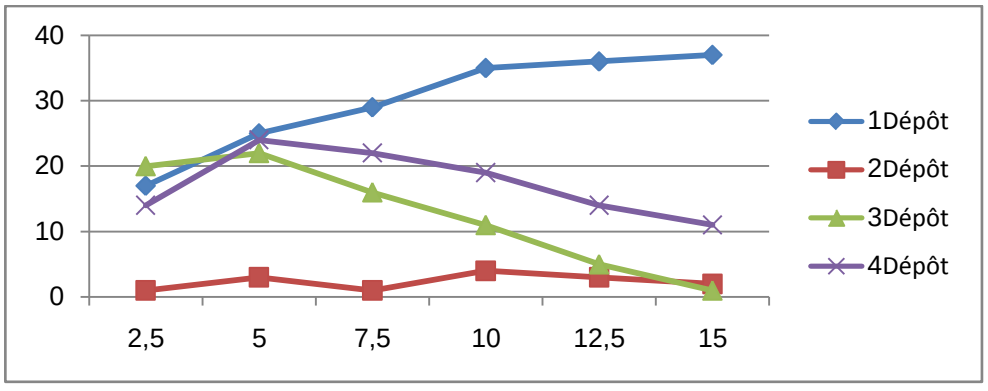


Figure 3.17. Influence du nombre des robots sur les performances (6 robots et 3 scouts)

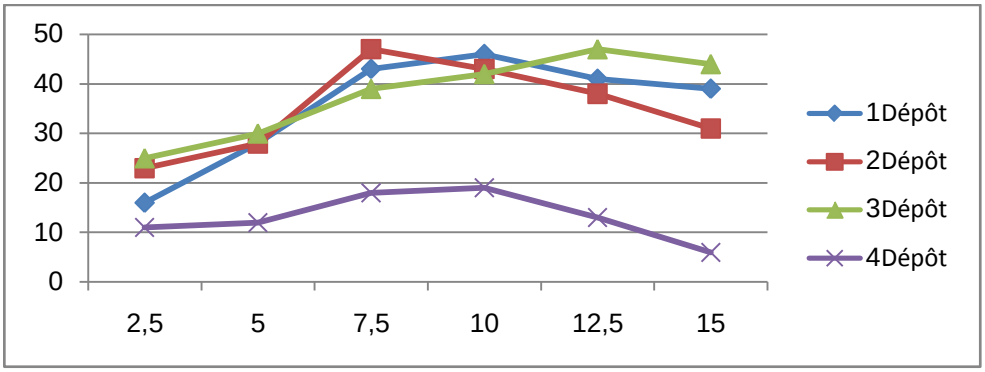


Figure 3.18. Influence du nombre des robots sur les performances
(8 robots et 4 scouts)

Discussion:

Le Tableau 3.14 et la Figure 3.6 montrent les résultats obtenus avec l'algorithme *AMB*, Le nombre d'objets augmente avec toute augmentation du nombre des robots, ce qui permet l'augmentation de nombre des objets dans les dépôts et aboutit à une satisfaction des demandes externe. Nous notons qu'une simulation avec 6 robots et 3 scouts a donné un résultat inattendu où la quantité des cibles n'a jamais dépassé 5 et le dépôt a raté 3 demandes, la cause de cette perturbation est le recrutement de tous les robots de ce dépôt avec un autre dépôt.

6. Conclusion

Dans ce chapitre nous avons présenté la plateforme ARGoS utilisé pour l'implémentation de notre algorithme. Nous avons présenté aussi les robots Foot-bot, ainsi que leurs caractéristiques. Ensuite, nous avons expliqué le comportement de notre algorithme *AMB*.

Nous avons ainsi analysé les performances de l'algorithme *AMB* pour satisfaire les demandes d'un système externe. De ce fait, nous avons proposé un ensemble de scénarios avec variation de certains paramètres pour voir s'ils ont une influence sur les performances. Les résultats montrent toujours l'efficacité de l'algorithme proposé.

L'algorithme proposé souffre de quelques inconvénients qui restent à investiguer dans le future proche. Ces inconvénients se résument aux points suivants : (1) quand les clusters sont positionnés dans des endroits séparés qui ne permettent pas une rencontre de robots, le processus de recrutement devient très difficile voire impossible dans certaines simulations, (2) quand un dépôt récupère sa stabilité, les robots qui sont déjà en dehors du dépôt avec un état en alerte, restent encore en alerte et donc continuer à recruter d'autres robots (même ceux qui connaissent déjà que le dépôt est stable).

CONCLUSION GÉNERALE

Conclusion générale

Le foraging à un lieu central (Central Place Foaging- CPF) est un type particulier de foraging dans lequel les robots doivent collecter des objets en un lieu central. Empruntant la terminologie à la biologie, la place centrale est aussi appelée *nid*. De nouveaux travaux du foraging ont proposé une nouvelle extension du foraging à plusieurs dépôts (Multi-Place Foraging- MPF). Le MPF utilise plusieurs dépôts au lieu d'un seul dépôt central. Les travaux les plus récents du foraging tentent de rapprocher le maximum possible des applications du monde réel en considérant certaines caractéristiques importantes qui reflètent les exigences des applications du monde réel tel que : l''adaptation en ligne, la durabilité du système et l'adaptabilité dans des scénarios critiques.

Dans ce travail, nous avons proposé un algorithme de foraging multi-dépots bio-inspiré de comportement des abeilles et la marche en Lévy-walk, où les robots scouts déplacent en Levy-walk pour rechercher des sources d'objets afin de permettre les robots employeurs de les récoltés. L'objectif principal de notre approche est d'assurer la satisfaction des demandes qui proviennent de l'extérieure. Les robots sont donc chargés de maintenir la durabilité ou la survie du système foraging en maintenant des quantités d'objets minimales tout le long de la vie du système. D'un coup, la stratégie d'exploration basé abeille peut être suffisante pour maintenir cette survie, en cas d'alerte, d'autre robots doivent être recrutés pour maintenir la survie du dépôt en alerte.

L'algorithme proposé *AMB* a été implémenté sus la plateforme de robotique mobile ARGoS, a été testé dans différentes configurations environnementales. Les résultats de simulation obtenus montrent que l'algorithme *AMB* est robuste, flexible et permet de maintenir la survie du système.

Enfin notre travail ouvre à la fois des perspectives qui se situent sur deux plans:

1. Un plan d'élargissement consiste à Etendre l'algorithme *AMB* pour prendre en charge le problème de saturation des dépôts et appliquer réellement l'algorithme proposé.
2. Un autre plan d'approfondissement qui permet de prendre en considération la quantité et la qualité des sources au niveau choix des sources par les robots employeurs.

BIBLIOGRAPHIES

Bibliographies

[Arthur and Vassilvitskii, 2007]*Arthur, D. &Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, SODA '07 (pp. 1027–1035). Philadelphia, PA: Society for Industrial and Applied Mathematics.*

[Banerjee and Moses, 2010b]*Banerjee, S., & Moses, M. (2010b). Scale invariance of immune system response rates and times: Perspectives on immune system architecture and implications for artificial immune systems. Swarm Intelligence, 4(4), 301–318.*

[Beekman et al., 2008] *Beekman, M., Sword, G. A. et Simpson, S. J. (2008). Biological foundations of swarm intelligence. In Blum, C. et Merkle, D., ´editeurs : Swarm Intelligence, Natural Computing Series, page 2–41. Springer-Verlag.*

[Berthold et Querner, 1981] *Berthold, P. et Querner, U. (1981). Genetic basis of migratory behaviour in european warblers. Science, 212:77–79.*

[Blum et Merkle, 2008] *Blum, C. et Merkle, D., ´editeurs (2008). Swarm Intelligence. Natural ComputingSeries. Springer-Verlag.*

[Bonabeau et al., 1999a] *Bonabeau, E., Dorigo, M. et Theraulaz, G. (1999a). Swarm Intelligence. From Natural to Artificial Systems, chapitre 1, pages 7–8. Studies in the Sciences of Complexity. Oxford University Press, New York Oxford.*

[Camazine et al., 2001] *Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G. etBonabeau, E. (2001). Self-Organization in Biological Systems. Princeton studies in complexity. Princeton University Press.*

[Chapman et al., 1989]*Chapman, C. A., Chapman, L. J.,&McLaughlin, R. L. (1989). Multiple central place foraging by spider monkeys: Travel consequences of using many sleeping sites. Oecologia, 79(4), 506–511.*

[Charrier, 2009]*Charrier Rodolphe. L'intelligence en essaim sous l'angle des systèmes complexes : étude d'un système multi-agent réactif à base d'itérations logistiques couplées. Modélisation et simulation. Université Nancy II, 2009. Français.*

[Cordon et al., 2002] *O.Cordon, F. Herrera, and T. Stutzle, " A review on the ant colony optimization metaheuristic: Basis, models and new trends ". Mathware and Soft Computing, 9(2–3),(2002), pp 141–175.*

[Deneubourg et al., 1990] *J.-L. Deneubourg, S. Goss, N.R. Franks, A. Sendova-Franks,C. Detrain, et L. Chretien. "'The dynamics of collective sorting: robot-like ant and ant-like robots"'.In Proceedings of the First International Conference on Simulation of Adaptive Behavior, pp 356–365, 1990.*

[Dorigoet al., 1991] *A. Colorni, M. Dorigo, V. Maniezzo , "'Distributed optimisation by ant colonies"', In Proceeding of ECAL-91,Paris, France 134-142, 1991.*

[Dorigoet al., 1996] *M. Dorigo, V. Maniezzo, A. Colorni " "' The Ant System: Optimization by a Colony of Cooperating Agents"', IEEE Transactions on Systems, Man and Cybernetics-Part B, 1996, 1(26): pp. 29-41.*

[Dorigo and Gambardella, 1997] *M. Dorigo and L. M. Gambardella ."'Ant Colony System: A cooperating learning approach to the travelling salesman problem"'. IEEE Transactions on Evolutionary Computation, 1(1),1997, pp 53–66.*

[Ericksen et al., 2017]John Ericksen, Melanie Moses Stephanie Forrest, Automatically Evolving a General Controller for Robot Swarms .978-1-5386-2726-6/17/$31.00 ©2017 IEEE

[Flanagan et al.2013].*Flanagan, T. P., Pinter-Wollman, N. M., Moses, M. E., & Gordon, D. M. (2013). Fast and flexible: Argentine ants recruit from nearby trails. PLOS ONE, 8(8), 1–7.*

[Hecker and Moses, 2013]*J. P. Hecker, and M. E. Moses, An Evolutionary Approach for Robust Adaptation of Robot Behavior to Sensor Error, Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO '13 Companion, ACM, no. 8, pp. 1437-1444, New York, NY, USA, 2013*

[Hecker and Moses, 2015] *Hecker. J. P and Moses. M. E, Beyond pheromones: evolving errortolerant, flexible, and scalable ant-inspired robot swarms, Swarm Intelligence, Springer US, vol. 9, pp. 43-70, 2015.*

[lu et al., 2016a]*Lu, Q., Hecker, J. P., & Moses, E. M. (2016a). The MPFA: A multipleplace foraging algorithm for biologically-inspired robot swarms. In IEEE/RSJ international conference on intelligent robots and systems (IROS 2016).*

[lu et al., 2016b] *Lu, Q., Moses, M., &Hecker, J. (2016b). A scalable and adaptable multiple-place foraging algorithm for ant-inspired robot swarms.*

[lu et al., 2018] *Qi Lu · Joshua P. Hecker · Melanie E. Moses. Multiple-place swarm foraging with dynamic depots. Article in Autonomous Robots · January 2018. DOI: 10.1007/s10514-017-9693-2*

[Matthew et al, 2016] *G Matthew Fricke, Joshua P Hecker, Antonio D Griego, Linh T Tran, and Melanie E Moses.A distributed deterministic spiral search algorithm for swarms.In Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, pages 4430-4436.IEEE, 2016.*

[Mayeur, 2014]*Bernard Mayeur. Outil automatique de placement de robots pour des expériences de SwarmRobotics.Mémoire présenté en vue de l'obtention Du diplôme du Master en Sciences informatiques Année académique 2013 – 2014. UNIVERSITE LIBRE DE BRUXELLES, UNIVERSITE D'EUROPE Faculté des Sciences Département d'Informatique*

[Nembrini and Martinoli, 2017] *Nembrini, J. &Martinoli, A. (2007). Robotique en essaim: récents résultats et directions futures. In Proc of the Journées Nationales de la Recherche en Robotique 2007 (No. EPFL-CONF-166168).*

[Pinciroli, 2011] *Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, Timothy Stirling, _Alvaro Guti_errez, Luca Maria Gambardella, and Marco Dorigo. ARGoS : a modular, multi-enginesimulator for heterogeneous swarm robotics. In Proceedings of theIEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011), pages 5027{5034. IEEE Computer Society Press, Los Alamitos, CA, September 2011.*

[Pinciroli, 2012] *Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, Mauro Birattari, Luca Maria Gambardella, and Marco Dorigo. ARGoS : a modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intelligence, 6(4) :271295, 2012.*

[Stanley and Millkkulainen, 2002] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," Evolutionary computation, vol. 10, no. 2, pp. 99–127, 2002

[Yazdani, 2017] *YazdaniHasan.Swarms Robots and their applications. IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 1, Ver. II (Jan.-Feb. 2017), PP 46-47. DOI: 10.9790/0661-1901024647.*

[Zedadra, 2015] *Ouarda ZEDADRA, Résolution des problèmes de coordination d'agents réactifs dans un environnement incertain, Thèse Pour obtenir le diplôme de Doctorat en sciences, Année 2015/2016*