

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 collaborative de foraging  
multi-robots**

---

---

**Encadré Par :**

**Dr. ZEDADRA Ouarda**

**Présenté par :**

**Benkirat Abderrahmane**

**Juillet 2019**

# REMERCIEMENTS

---

## ***Je tiens tout d'abord***

à remercier Dieu le tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce Modeste travail.

## ***En second lieu,***

Je tiens à remercier mon encadreur Dr. ZEDADRA Ouarda, enseignante à l'Université de GUELMA, pour tout le soutien, l'aide, l'orientation, la guidance, ainsi que pour ses précieux conseils et ses encouragements lors de la réalisation de mon mémoire. J'ai beaucoup appris à son contact et ce fut un grand plaisir de travailler avec elle. Je voudrai lui adresser mes vifs remerciements et de lui témoigner mon sincère reconnaissance.

## ***Je tiens ensuite***

à remercier mes parents pour le soutien inconditionnel dont ils ont fait preuve depuis que mon projet est défini. Merci pour le soutien financier, moral, psychologique et matériel.

## ***Je remercie***

mes amis et camarades de la promotion **2019** pour ces cinq années passées ensemble, dans les meilleurs moments comme dans les pires.

## ***Afin de n'oublier personne,***

mes vifs remerciements s'adressent à tous ceux qui m'ont aidé à la réalisation de ce modeste mémoire.

# DEDICACE

---

Je tiens en tout premier lieu à remercier **Allah**.

Je voudrais dédier ce travail :

**A ma très chère mère Akila**

*Aucune dédicace ne saurait être assez éloquente pour Exprimer ce que tu mérites pour tous les sacrifices que tu n'as Cessé de me donner depuis ma naissance, durant mon enfance Et même à l'âge adulte.*

**A mon très cher père Hafid**

*Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être.*

**A mes beaux frères**

*Abdelwaheb et yahia. Que Dieu les bénisse pour moi.*

**A mon Encadreur ZEDADRA Ouarda.**

**A mes camarades et Mes collègues**

*Ibtissem Bouafia et Fethia Mesbah Qui m'ont aidé dans la réalisation de ce travail*

**Et à tous ceux que j'aime et à toutes les personnes**

*Qui m'ont prodigué des encouragements et se sont donné la peine de me soutenir durant cette formation.*

**Enfin je le dédie aux enseignants et ma promo et mes collègues d'informatiques.**

*Benkirat Abderrahmene*

## Résumé

Le problème de foraging constitue un benchmark dans la robotique mobile. Dans lequel les robots sont chargés de chercher, collecter et transporter un ou plusieurs objets à un ou plusieurs dépôts. Un foraging dans lequel les objets sont déposés dans un seul dépôt central s'appelle foraging centralisé (Central Place Foraging- CPF). Ce dernier, a été ignoré au fil du temps à cause des collisions créées par les robots au niveau du dépôt central. Pour surmonter ce dernier problème, la communauté a plutôt orienté vers l'utilisation de plusieurs dépôts créant alors le foraging multi-dépôt (Multi-Place Foraging- MPF).

Les travaux existants sur le foraging négligent certaines caractéristiques très importantes pour un bon déploiement des systèmes dans le monde réel tel que : l'adaptabilité en ligne, la durabilité..., etc.

Une nouvelle branche de foraging est alors créée dans l'esprit de rapprocher le maximum possible les systèmes de foraging aux applications du monde réel, nommé foraging durable. Dans une telle branche, le système tente de répondre à des exigences de systèmes externes.

Dans ce mémoire, nous avons proposé un algorithme de foraging durable multi-dépôts dans lequel les robots tentent de satisfaire les demandes d'un système externe. Notre proposition fait une hybridation entre les algorithmes bio-inspirés Lévy walk et Firefly dont l'objectif est de combiner les avantages des deux algorithmes afin d'assurer la survie de dépôts pendant la durée de la mission de foraging. L'algorithme proposé *Lévy walk Firefly Recruiting Algorithm (LWFRA)* utilise le Lévy walk pour l'exploration de l'environnement, et le Firefly pour recruter des robots en cas de nécessité. Les tests expérimentaux montrent la supériorité de l'algorithme *LWFRA* dans différentes configurations.

**Mots clé :** Robotique en essaim, Foraging durable, Foraging multi-dépôts, Lévy Walk, FireFly .

# Sommaire

Résumé .....	i
Sommaire.....	1
Liste des figures.....	3
Liste des tableaux.....	5
Liste des équations.....	6
Liste des algorithmes.....	7
Introduction générale.....	8

## Chapitre 01 : Foraging Multi-Robots- définitions

### et travaux reliés

Introduction.....	10
1. Intelligence en essaim.....	11
2. Robotique en essaim.....	12
2.1. Définitions .....	12
2.2. Comportements de la robotique en essaim.....	13
2.2.1. Comportements de base.....	13
2.2.2. Comportement combiné.....	14
3. Le problème de foraging.....	17
3.1. Définitions.....	17
3.2. Comportement de référence d'un Foraging robot.....	17
3.3. Types de foraging.....	18
3.4. La taxonomie des travaux reliés.....	19
3.5. Le foraging centralisé : travaux reliées.....	20
3.6. Le foraging multi-dépôts : travaux reliées.....	24
4. Conclusion.....	29

## Chapitre 02 : Conception

1. Introduction.....	30
2. Problématique.....	30
3. Objectifs et Propositions.....	31

<b>4. Analyse des Algorithmes reliés.....</b>	<b>32</b>
4.1. Algorithme Lévy Walk.....	32
4.2. Algorithme Firefly.....	33
<b>5. Algorithme proposé : Lévy Walk Firefly Recruiting Algorithm .....</b>	<b>34</b>
5.1. Structure paramétrique de LWFRA .....	34
5.2. Comportement des robots LWFRA .....	35
5.2.1. Description textuelle des différents états.....	36
5.2.2. Organigramme de LWFRA.....	37
5.2.3. Pseudo code de l'algorithme LWFRA .....	38
<b>6. Conclusion .....</b>	<b>40</b>

## **Chapitre 03 : Implémentation et Expérimentations**

<b>1. Introduction.....</b>	<b>42</b>
<b>2. Environnement de développement Netlogo.....</b>	<b>42</b>
<b>3. Présentation générale de l'application.....</b>	<b>44</b>
3.1. Interface graphique du système.....	44
3.2. Les composants de notre système et leur Caractéristiques.....	45
<b>4. Etude de paramètres de l'algorithme LWFR.....</b>	<b>46</b>
<b>5. Simulation et analyse des résultats.....</b>	<b>47</b>
5.1. Critères de performances.....	47
5.2. Scénarios de simulation.....	47
5.3. Résultats, discussions et comparaisons .....	48
5.3.1. Scénario 1.....	48
5.3.2. Scénario 2.....	49
5.3.3. Scénario 3.....	51
<b>6. Conclusion.....</b>	<b>53</b>

<b>Conclusion générale et perspectives.....</b>	<b>54</b>
---	-----------

<b>Bibliographie.....</b>	<b>55</b>
---------------------------	-----------

## Liste des Figures

### Chapitre 01 : Foraging Multi-Robots- définitions et travaux reliés

<b>Figure 1.1</b> : Exemples d'intelligence en essaim.....	12
<b>Figure 1.2</b> : Dispersion [Schwager et al.,2006].....	13
<b>Figure 1.3</b> : Agrégation [Li et al.,2016].....	14
<b>Figure 1.4</b> : Mouvement collectif [Baldassarre <i>et al.</i> , 2007].....	14
<b>Figure 1.5</b> : Exploration [Gasparri et al., 2012].....	15
<b>Figure 1.6</b> : Couverture [Correll et al., 2008].....	15
<b>Figure 1.7</b> : Localisation des cibles [Hayes <i>et al.</i> , 2002].....	16
<b>Figure 1.8</b> : Manipulation collective [Martinoli, 1999].....	16
<b>Figure 1.9</b> : Fourragement [Krieger & Billeter, 2000].....	16
<b>Figure 1.10</b> : Machine d'état finis pour le Foraging collective [Winfield et al., 2010].....	17
<b>Figure 1.11</b> : Taxonomie de Foraging Multi-Robots.....	20
<b>Figure 1.12</b> : Essaim de robot iAnt autour d'un dépôt éclairé [Hecker & Moses,2015].....	21
<b>Figure 1.13</b> : Comportements des robots [Hecker & Moses,2015].....	21
<b>Figure 1.14</b> :Environnement de simulation <i>Considéré par Castello et al.</i> [Castello et al.,2013].....	22
<b>Figure 1.15</b> : Comportement des robots [Castello et al., 2014].....	23
<b>Figure 1.16</b> : L'environnement considéré dans [Castello et al., 2016].....	24
<b>Figure 1.17</b> : Infrastructure ROS [Castello et al., 2016].....	24
<b>Figure 1.18</b> : Comportement des robots MPFA [Lu et al., 2018].....	26
<b>Figure 1.19</b> : Les différents environnements utilisés dans le travail proposé [Lu et al., 2018] .....	27

### Chapitre 02 : Conception

<b>Figure 2.1</b> : Exemple de Lévy walk [Reynolds A.M, 2018].....	32
<b>Figure 2.2</b> : Luciole « FireFly ».....	33
<b>Figure 2.3</b> : Organigramme représentant le comportement des robots LWFRA.....	38

### Chapitre 03 : Implémentation et Expérimentations

<b>Figure 3.1</b> : Plateforme Netlogo.....	43
<b>Figure 3.2</b> : Interface graphique de l'application.....	45
<b>Figure 3.3</b> : Comparaison entre la distribution de la taille des pas .....	46
<b>Figure 3.4</b> : Influence de nombre de dépôts sur la survie global.....	48

## Liste des Figures

---

<b>Figure 3.5</b> : Influence de la taille de l'environnement sur le taux de survie .....	49
<b>Figure 3.6</b> : Influence de la taille de l'environnement sur le taux de survie.....	50
<b>Figure 3.7</b> : Influence de la taille de l'environnement sur la survie global.....	50
<b>Figure 3.8</b> : Influence de nombre de robots sur le taux de survie.....	51
<b>Figure 3.9</b> : Influence de nombre de robots sur le taux de survie.....	52
<b>Figure 3.10</b> : Influence de nombre de robots sur la survie global.....	52



# Liste des Tableaux

## **Chapitre 01 : Foraging Multi-Robots- définitions et travaux liés**

**Tableau 1.1** : Comparaison qualitative des travaux liés.....28

## **Chapitre 02 : Conception**

**Tableau 2.1** : Abréviations utilisées dans l’algorithme LWFRA.....35

## **Chapitre 03 : Implémentation et Expérimentations**

**Tableau 3.1** : Les trois scénarios de simulation.....47

**Tableau 3.2** : Résultats du scénario 1.....48

**Tableau 3.3** : Influence de la taille de l’environnement sur les performances .....49

**Tableau 3.4** : Influence de la variation de nombre de robots sur les performances.....51

## Liste des équations

<b>Equation (1)</b> : La distribution de probabilité de Lévy.....	32
<b>Equation (2)</b> : La distribution de probabilité de Lévy approximé.....	32
<b>Equation (3)</b> : Générateur de longueur des pas .....	32
<b>Equation (4)</b> : Intensité lumineuse .....	33
<b>Equation (5)</b> : Equation d'attractivité.....	33
<b>Equation (6)</b> : Equation de distance entre deux lucioles.....	34
<b>Equation (7)</b> : Equation de mouvement d'une luciole.....	34
<b>Equation (8)</b> : Générateur des pas aléatoire de Lévy Walk.....	34
<b>Equation (9)</b> : Equation d'intensité en fonctions des cibles .....	34
<b>Equation (10)</b> : Fonction de réduction de l'intensité lumineuse .....	34
<b>Equation (11)</b> : Equation de déplacement des robots LWFR.....	35
<b>Equation (12)</b> : Fonction d'augmentation de la vitesse des robots LWFR .....	35
<b>Equation (13)</b> : Equation de communication entre les robots LWFR .....	35
<b>Equation (14)</b> : La différence entre la quantité d'objet.....	36
<b>Equation (15)</b> : Fonction de décision.....	36
<b>Equation (16)</b> : Le taux de survie de dépôt .....	47
<b>Equation (17)</b> : Fonction caractéristiques.....	47
<b>Equation(18)</b> : La survie globale de système .....	47

# Liste des algorithmes

## Chapitre 02 : Conception

Algorithme 2.1 : Robot en attente.....	39
Algorithme 2.2 : Chercher.....	39
Algorithme 2.3 : Collecter.....	39
Algorithme 2.4 : Recruter.....	40

# Introduction générale

## 1. Positionnement scientifique

La notion d'intelligence collective est née dans les années 60, de l'application, à la robotique, des analyses des éthologues sur le comportement des sociétés d'insectes. Aujourd'hui, les modèles de comportements collectifs des insectes sociaux fournissent aux informaticiens et roboticiens des méthodes puissantes pour la conception d'algorithmes d'optimisation et de contrôle distribué. L'intelligence en essaim a pour objet de transformer les capacités collectives de résolution de problèmes des insectes sociaux en techniques artificielles de résolution de problèmes scientifiques. L'intelligence en essaim est utilisée pour différentes applications : les animations graphiques, les algorithmes d'optimisation et la robotique collective. Ce mémoire traitera exclusivement de la robotique collective, appelée robotique en essaim dans laquelle les principes de l'intelligence en essaim sont appliqués.

La robotique en essaim est un champ de recherche qui s'intéresse à la coordination auto-organisée de systèmes multi-robots, et tout particulièrement à ceux qui consistent en un grand nombre d'unités dont la complexité est minimisée. Originellement inspiré par des exemples naturels, principalement dus aux insectes sociaux. Les robots peuvent être utilisés pour résoudre différents problèmes, par exemple, l'agrégation d'objets, la ségrégation, l'exploration, les manipulations coopératives, le foraging..., etc.

Le foraging Multi-Robots forme une instance de plusieurs applications du monde réel comme : le nettoyage, le déminage, l'exploration des environnements hostiles, la recherche et le sauvetage. Le foraging est soit centralisé, connu aussi sous le nom Central Place Foraging (CPF), quand les robots délivrent les objets à un seul dépôt central, soit multi-dépôts connus aussi sous le nom Multi-Place Foraging (MPF), quand les robots délivrent les objets à plusieurs dépôts. Les MPF viennent pour surmonter les problèmes rencontrés dans le CPF à savoir : les collisions autour du dépôt central, et les longs chemins de trajets entre les sources et le dépôt central. On s'intéresse dans notre travail au foraging Multi-dépôts.

## 2. Problématique et objectifs

Malgré les travaux réalisés au cours des années, aucun système de foraging n'a été déployé dans le monde réel, voire aucun n'a été projeté sur une application similaire. Cela revient au fait que ces travaux n'ont pas considéré 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. Nous cherchons donc à considérer ces nouvelles caractéristiques en répondant aux questions suivantes :

1. Est-il possible de maintenir la durabilité du système de foraging, autrement dit, est-il possible de satisfaire toutes les demandes qui proviennent de l'extérieure à travers une collaboration entre robots appartenant à des dépôts étrangers.
2. Quel modèle d'interaction faut-il utiliser pour garantir une meilleure collaboration qui ne demande pas de communication coûteuse ?

Pour répondre à ces questions nous avons :

1. Proposé un système de foraging multi-dépôt qui permet de satisfaire les demandes qui proviennent de l'extérieur. Les robots dans ce système utilisent un algorithme bio-inspiré Lévy Walk pour l'exploration et la recherche des objets ;
2. Si un dépôt est en alerte, ses robots doivent recruter d'autres robots en utilisant une attraction lumineuse basée sur l'algorithme Firefly.

### 3. Plan de mémoire

Ce manuscrit est réparti en trois chapitres :

**Le premier chapitre** : dans ce chapitre, nous présentons les domaines de recherche qui sont l'intelligence en essaim et la robotique en essaim. Ensuite, nous focalisons sur le problème de foraging où nous présentons quelques définitions et concepts reliées. Puis, nous synthétisons les travaux de foraging reliés, et nous comparons ces travaux dans un tableau avec un ensemble de critères que nous avons jugé utiles.

**Le deuxième chapitre** : ce chapitre est consacré à la conception du problème. Nous présentons donc la problématique de recherche, les objectifs et la proposition. Nous focalisons par la suite, sur l'algorithme proposé *LWFR* où nous présentons l'organigramme de ce dernier, une description textuelle des états et les pseudos codes associés.

**Le troisième chapitre** : Nous présentons dans ce chapitre les résultats de mise en œuvre et de test de l'algorithme proposé. Une présentation de la plateforme de simulation Netlogo, les caractéristiques des robots utilisés (agents) est aussi faite. Le présent manuscrit est clôturé par une conclusion récapitulative et quelques perspectives à explorer en continuité de mémoire.

# Chapitre 01 : Foraging Multi-Robots- définitions et travaux reliés

## Introduction

L'intelligence en essaim est une discipline d'intelligence artificielle moderne qui concerne la conception de systèmes multi-agents avec des applications, telles que l'optimisation et la robotique. Le paradigme de conception de ces systèmes est fondamentalement différent des approches les plus traditionnelles. L'inspiration pour la conception de ces systèmes provient du comportement collectif d'insectes sociaux tels que fourmis, termites, abeilles et guêpes, ainsi que du comportement d'autres sociétés animales telles que des groupes d'oiseaux ou des bancs de poissons. Une des branches les plus actives de l'intelligence en essaim est la robotique en essaim. C'est l'étude de la façon de faire collaborer des robots par l'application des méthodes de l'intelligence en essaim pour résoudre une tâche, qui serait autrement impossible à résoudre par un seul robot. Le foraging multi-robots est considéré comme un benchmark dans la robotique mobile. Il constitue une métaphore de plusieurs problèmes du monde réel. Pour se rapprocher le maximum possible des problèmes du monde réel, les travaux récents sur le problème de foraging multi-robots considèrent une nouvelle extension qui est le foraging durable. Dans lequel, les robots doivent répondre à des demandes qui viennent de l'extérieure et d'assurer ainsi la survie de leurs dépôts.

Notre objectif dans ce mémoire, c'est de considérer cette nouvelle extension du foraging. Nous cherchons donc à proposer un algorithme de foraging qui répond aux exigences d'un système extérieur.

Ce chapitre se trouve divisé en quatre parties principales. Dans la première, nous présentons le domaine d'inspiration qui est l'intelligence en essaim avec quelques définitions. Tandis que, dans la deuxième partie, nous présentons la robotique en essaim avec des définitions et quelques domaines d'applications. La troisième partie focalise sur le problème de foraging où nous présentons quelques définitions et concepts reliées. La dernière partie est consacrée à la synthèse des travaux reliés.

### 1. Intelligence en essaim

Le terme d'intelligence en essaim a été initialement défini et utilisé dans l'article de Beni et Wang [Beni et Wang, 1989] intitulé "Swarm intelligence in cellular robotics systems". Cet article fondateur du concept proposait une alternative à l'expression de robotique cellulaire pour décrire un groupe de robots coopérant pour accomplir une tâche ou résoudre un problème.

Beni caractérise la **notion d'essaim** par : « *un groupe constitué d'un nombre d'au moins une centaine d'entités, mais très inférieur aux populations de particules considérées en physique pour lesquelles des approches statistiques sont nécessaires. D'autre part, les entités du groupe sont simples dans leur comportement individuel, sont quasi-identiques, et le contrôle de l'essaim se réalise en mode décentralisé.* »

**Définition 1.1** : Beni [Beni, 2005], définit l'intelligence de l'essaim comme : « *une aptitude à générer des schémas ordonnés de façon imprévisible, schémas sur lesquels une analyse est faite par chaque robot pour optimiser si besoin ses actions en vue d'atteindre un objectif collectif fixé à l'avance* ».

Aujourd'hui, l'intelligence en essaim est un champ de recherche scientifique, multidisciplinaire, qui s'intéresse aux processus distribués (non supervisés) d'organisation et de résolution de problèmes présents dans un certain nombre de sociétés animales et dans des systèmes artificiels qui en sont inspirés [Garnier, 2008].

**Définition 1.2** : Merkle et Blum [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* ».

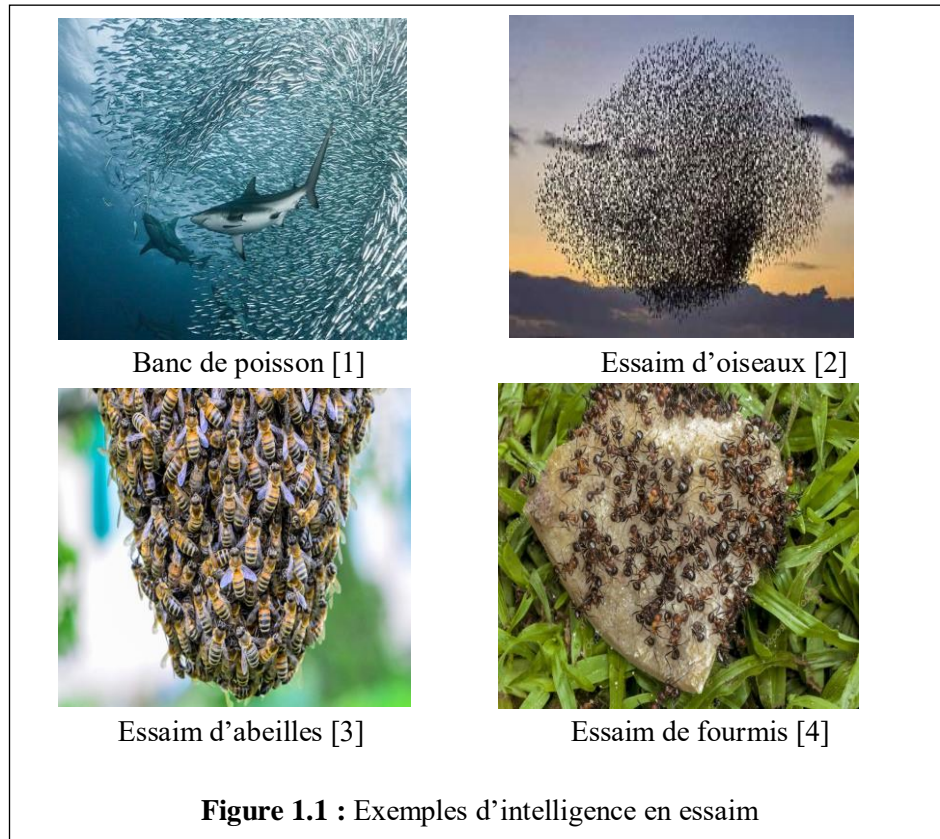
**Définition 1.3** : Charrier et al. [Charrier et al., 2008], propose une autre définition de l'intelligence en essaim en informatique « *l'intelligence artificielle en essaim recouvre un ensemble d'algorithmes utilisés en informatique qui ont la particularité de faire interagir de multiples processus élémentaires de façon décentralisée, c'est-à-dire sans contrôleur global, afin de résoudre des problèmes* ».

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

- Des unités simples sont plus faciles à produire en masse et sont interchangeables.

- Le système est plus fiable, plus robuste aux diverses défaillances.
- L'essaim est doté d'importantes capacités d'adaptation à son environnement.

La *Figure 1.1* montre quelque exemple de systèmes jugés d'intelligence en essaim



## 2. Robotique en essaim

### 2.1. Définitions

La robotique en essaim est actuellement l'une des applications primaires de l'intelligence en essaim. Les essaims fournissent la possibilité accrue de réalisation d'une tâche (tolérance aux erreurs), ainsi qu'une faible complexité des unités et enfin un coût faible au regard des systèmes robotiques traditionnels. Les domaines d'application de ces robots sont variés : vaisseaux spatiaux, inspection /maintenance, construction, agriculture ou médecine.

**Définition 1.4** : Beni [Beni, 2004], définit l'intelligence dans système de la robotique en essaim comme : « *une aptitude à générer des schémas ordonnés de façon imprévisible, schémas sur lesquels une analyse est faite par chaque robot pour optimiser si besoin ses actions en vue d'atteindre un objectif collectif fixé à l'avance.* »



**Définition 1.5** : Dorigo et Sahin [Dorigo et Sahin, 2004], le définit comme : « *la robotique en essaim est l'étude de la façon dont un grand nombre d'agents relativement simples incarnés physiquement peuvent être conçus pour qu'un comportement voulu puisse émerger des interactions locales entre agents et entre les agents et l'environnement* »

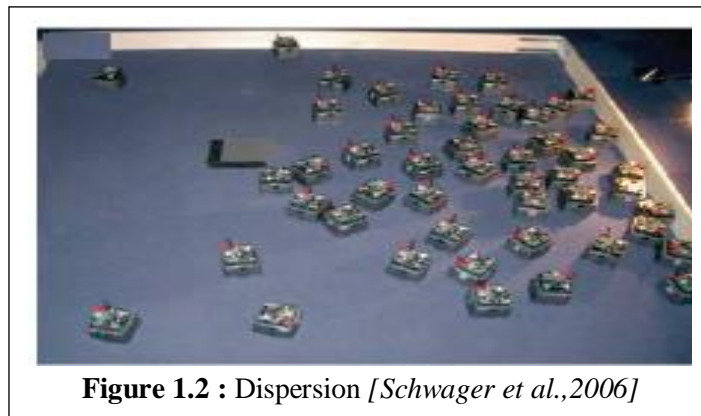
### 2.2. Comportements de la robotique en essaim

Il existe deux types de comportements dans la robotique en essaim : les comportements de base comme l'agrégation, et les comportements combinés comme le foraging.

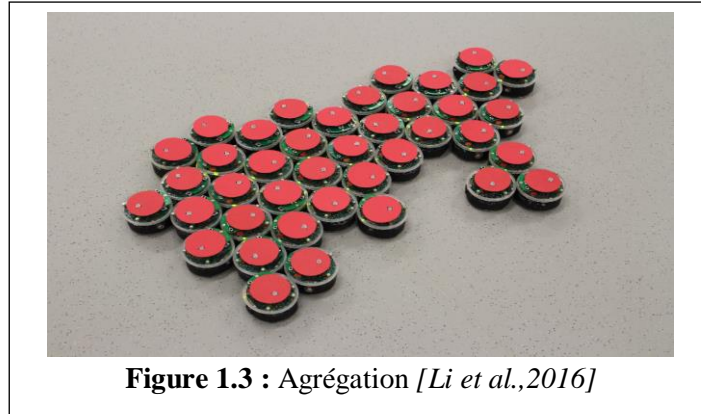
#### 2.2.1. Comportements de base

Les comportements de base sont des comportements simples qui sont à la base des comportements les plus complexes. Nous présentons dans ce qui suit certains de ces comportements.

- 1) **Dispersion** : La dispersion contrôlée dans un environnement est une composante comportementale importante pour un essaim de robots, de par le fait que cette dispersion peut servir par exemple de réseau de communication [McLurkin & Smith, 2007] voir de senseur robotique distribuée [Schwager et al., 2006]. Dans ce dernier exemple (Figure 1.2), Schwager et al. [Schwager et al., 2006] démontrent la capacité d'un essaim de 50 unités de se disperser de manière optimale pour échantillonner une fonction de densité spécifique.



- 2) **Agrégation** : Beaucoup d'applications de robotique en essaim demandent aux membres de l'essaim de se regrouper pour par exemple échanger des informations. Bien que beaucoup de recherches s'intéressent à l'agrégation, ces résultats sont principalement théoriques ou issus de simulations. La figure 1.3 montre une agrégation d'essaim de robot



3) **Mouvement collectif** : inspiré par le travail de *Reynolds* sur la modélisation des essaims d'oiseaux [Reynolds, 1987], le mouvement collectif est un autre comportement de base important. Dans le cadre du projet *SwarmBot* [Baldassarre et al., 2007] (Figure 1.4), il est présenté sous forme de mouvement collectif basé sur un essaim physiquement connecté, qui permet la traversée par le collectif de larges vides qu'un seul robot ne saurait être en mesure de franchir.



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.

### 2.2.2. 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. Dans cette partie nous présentons quelques comportements combinés de la robotique en essaim.

- 1) **Exploration:** L'exploration implique la nécessité pour un essaim de robots de découvrir l'environnement. La *Figure 1.5*, montre un essaim de robots dans la mission d'exploration.



**Figure 1.5 :** Exploration [Gasparri *et al.*, 2012]

- 2) **Couverture:** La couverture d'une région implique un passage systématique en tous les points. Cette tâche trouve son application par exemple dans les tondeuses à gazon.



**Figure 1.6 :** Couverture [Correll *et al.*, 2008]

- 3) **Localisation de cible:** Forme l'une des applications majeures de la robotique en essaim. Cette tâche peut avoir de nombreuses variantes, dépendantes de la nature de la cible et de la manière de la localiser.



Figure 1.7 : Localisation des cibles [Hayes *et al.*, 2002]

- 4) **Manipulation collective:** Les exemples de manipulation collective se présentent soit sous la forme d'un transport collectif d'un unique objet [Kube & Bonabeau, 2000], soit sous la forme d'une agrégation d'objets [Martinoli *et al.*, 1999]. Dans une expérience analogue, un essaim de robots est appelé à coopérer pour extraire des tiges de leurs emplacements [Ijspeert *et al.*, 2001].

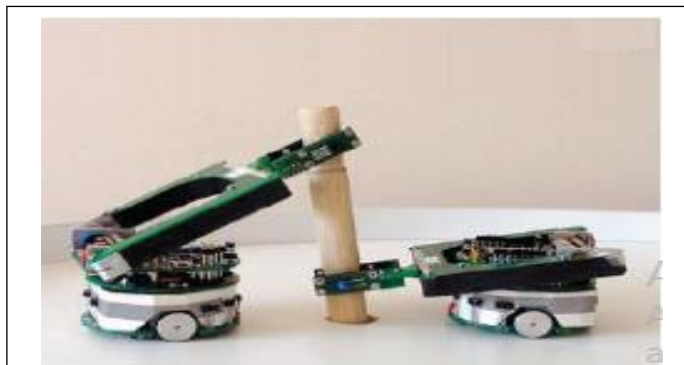


Figure 1.8 : Manipulation collective [Martinoli, 1999]

- 5) **Fourragement:** Analogue à la recherche de cible, la tâche de foraging, directement inspirée des fourmis, y ajoute la nécessité de ramener les cibles en un point. Krieger *et Billeter* présentent une étude impliquant un essaim de 12 robots, comparant le bénéfice de différents types de communication [Krieger & Billeter, 2000].

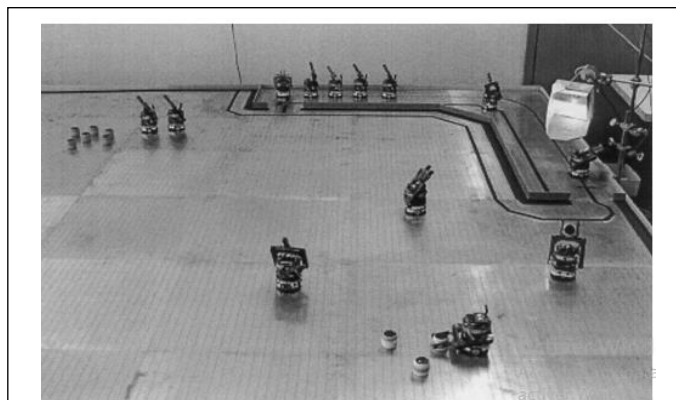


Figure 1.9 : Fourragement [Krieger & Billeter, 2000]

### 3. Le problème de Foraging

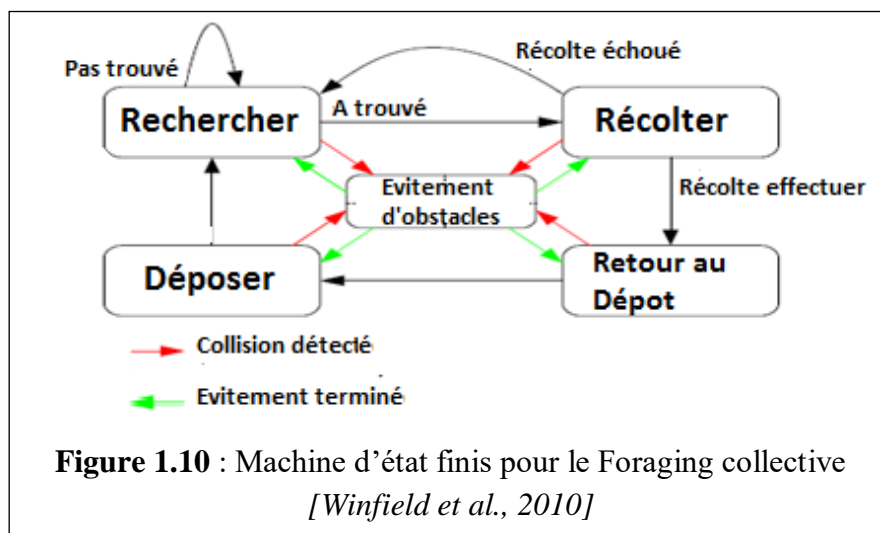
#### 3.1. Définition

Le Foraging, (La recherche de nourriture) est un problème de référence en robotique mobile. L'étude du foraging est importante pour plusieurs raisons: (1) parce que le foraging est une métaphore d'une vaste catégorie de problèmes ( le nettoyage, la récolte, la recherche et le sauvetage, le déminage ou l'astrobiologie planétaire) ; (2) c'est une tâche complexe qui inclut l'étude et la coordination de plusieurs autres tâches intégrant l'exploration, la navigation, la manipulation et le transport d'objets (3) aucun travail sur les robots fourrageurs n'a été déployé dans le monde réel, tous les travaux se trouvent encore dans les laboratoires de recherche pour des tests. [Winfield, 2009].

#### 3.2. Comportement de référence d'un Foraging robot

Les robots fourrageurs sont des robots mobiles capables de rechercher, transporter des objets trouvés vers un ou plusieurs points de collecte (dépot). Les robots peuvent être des robots uniques fonctionnant individuellement ou des robots multiples fonctionnant collectivement [Winfield, 2009]

La Figure 1.10 montre une représentation de la machine d'états finis (FSM) d'un robot fourrageur. Selon cette machine d'état, chaque robot se trouve toujours dans l'un des cinq états: *rechercher*, *récolter*, *retourner*, *déposer* ou *éviter les obstacles*.



Les cinq états sont définis comme suit [Winfield et al., 2010] :

- 1) **Rechercher**: Dans cet état, un robot se déplace physiquement dans l'espace de recherche en utilisant ses capteurs pour localiser et reconnaître les objets. Le robot adapte une marche aléatoire ou il utilise une stratégie systématique. Lorsque le robot

localise un élément, il change vers l'état *récolter*. Si le robot ne parvient pas à trouver la cible alors il reste dans l'état de recherche, la recherche est donc l'état par défaut.

- 2) **Récolter** : Dans cet état, un robot capture physiquement l'objet. Dès que l'objet a été récolté, le robot change à l'état *retour au dépôt* (Homing).
- 3) **Retour au dépôt (Homing)** : Dans cet état, un robot doit, avec son objet collecté, se déplacer vers un dépôt. Le Homing nécessite un certain nombre d'étapes, d'une part, la détermination de la position du dépôt par rapport au lieu où le robot se trouve à présent, d'autre part, l'orientation vers cette position et troisièmement, la navigation vers le dépôt. Une fois le robot a atteint le dépôt, il passe à l'état *déposer*.
- 4) **Déposer** : Dans cet état, le robot dépose ou délivre l'objet dans le dépôt, puis reprend immédiatement sa recherche.
- 5) **Evitement d'obstacles** : Dans un système de robotique mobile, les robots sont susceptibles de se heurter les uns aux autres, surtout si deux robots ou plus convergent vers le même objet. Ainsi, si des capteurs de prévention de collision de robot sont déclenchés lorsque le robot est dans l'un des quatre états ci-dessus, il doit alors prendre des mesures d'évitement avant de revenir dans cet état.

### 3.3. Types de Foraging

Dans les systèmes classiques de foraging, les robots transportent les objets collectés dans un seul dépôt central 'CPF', ou ils les transportent vers plusieurs dépôts 'MPF'. Le foraging dans sa version récente répond à des exigences de systèmes externes, il est connu sous le foraging durable.

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 et reviennent à ce dépôt central pour déposer ces ressources. Le foraging centralisé peut être instancié en un certain nombre d'applications de collecte de ressources réels, notamment la récolte et la collection de ressources extra-planétaires.

A cause de la surpopulation autour du dépôt central, les collisions augmentent avec l'augmentation du 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.

Le foraging multi-dépôts (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.

### 3.4. La taxonomie des travaux reliés

La *Figure 1.11* montre notre taxonomie pour bien classer les travaux reliés du foraging multi-robots (quelques travaux récents). La classification des travaux a été faite selon deux critères, l’instance et le type de foraging.

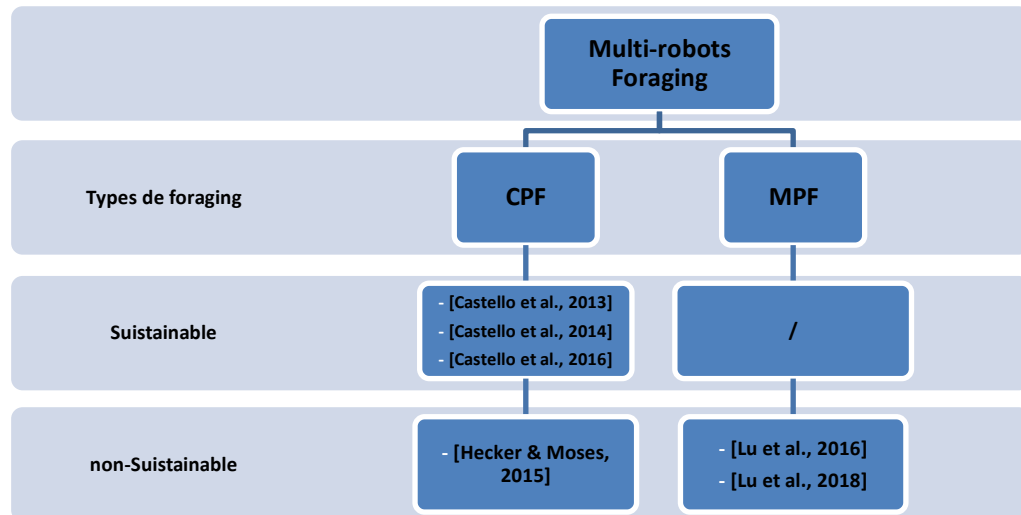


Figure 1.11: Taxonomie de Foraging Multi-Robots

### 3.5. Le foraging centralisé : travaux reliés

Le foraging centralisé a été considéré largement dans la littérature. Nous présentons dans cette section les travaux récents et pertinents du foraging centralisé.

#### 1) *Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms [Hecker et Moses,2015]*

Hecker et Moses [Hecker et Moses,2015] décrivent un système de robotique en essaim qui simule les comportements des fourmis. Ces comportements à des stratégies de recherche de nourriture qui optimisent les performances dans un environnement précise pour la mission de foraging. Le système proposé offre plusieurs solutions pour les problèmes rencontrés dans l'environnement du travail, parmi ces solutions :

- Augmente la communication lorsque l'information capturée est fiable et les ressources sont groupées (clusters).
- Moins de communication et plus de mémoire individuelle lorsque la taille des clusters est variable.
- Une grande dispersion lorsque la taille de l'essaim augmente.

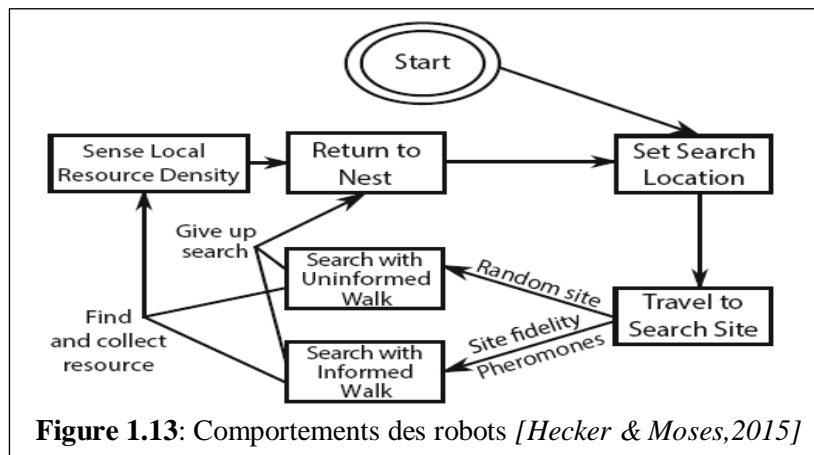
Les auteurs proposent un algorithme de foraging centralisé qui simule le comportement de foraging des fourmis « seed-harvester ». Ils utilisent un algorithme génétique (GA) pour améliorer les performances de l'approche proposé, cette approche a été testé sur des robots réels 'iAnt robot' (voir Figure 1.12) ainsi que par des simulations informatiques.





**Figure 1.12** : Essaim de robot iAnt autour d'un dépôt éclairé [Hecker & Moses,2015]

Au démarrage de la mission de foraging, chaque robot se trouve dans le dépôt. Il choisit une destination et se déplace à cette dernière. Ensuite, il commence sa recherche jusqu'à ce qu'il trouve un objet, il le collecte et perçoit la densité de ressources ou il se trouve. Finalement, il rentre au dépôt central pour déposer l'objet (voir *Figure 1.13*). Cette approche permet à un essaim de robots capable de fonctionner dans les environnements réel complexe.



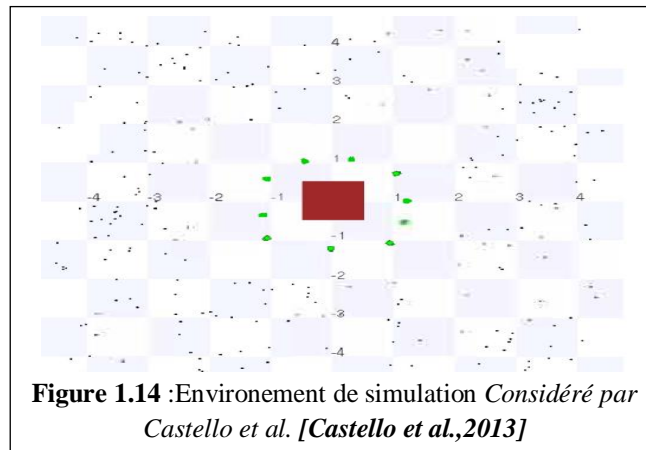
**Figure 1.13**: Comportements des robots [Hecker & Moses,2015]

### 2) *Task Allocation for a Robotic Swarm Based on an Adaptive Response Threshold Model* [Castello et al., 2013]

Castello et al [Castello et al., 2013], ont proposé un algorithme d'allocation des tâches optimisé pour le foraging centralisé (à un seul dépôt), afin d'avoir une division du travail efficace et adaptative pour un essaim de robots de petite taille (10 robots).

Les auteurs propose le model ARTM (Adaptive Response Threshold Model) [Yang et al., 2009], basé sur le modèle DASM (Discrete Attractor Selection Model) [Kashiwagi et

al.,2006] pour faire le calcul dynamique de quelques paramètres de foraging (paramètre de seuillage,...).L'environnement est considéré comme une zone rectangulaire de  $10 \times 10$  mètres avec un seul dépôt au centre, autour du dépôt il y a un ensemble de robots distribué aléatoirement et une certaine quantité de nourriture positionnée d'une manière aléatoire (voir *Figure 1.14*)



Le cycle d'allocation des tâches proposé est composé des 3 états suivants :

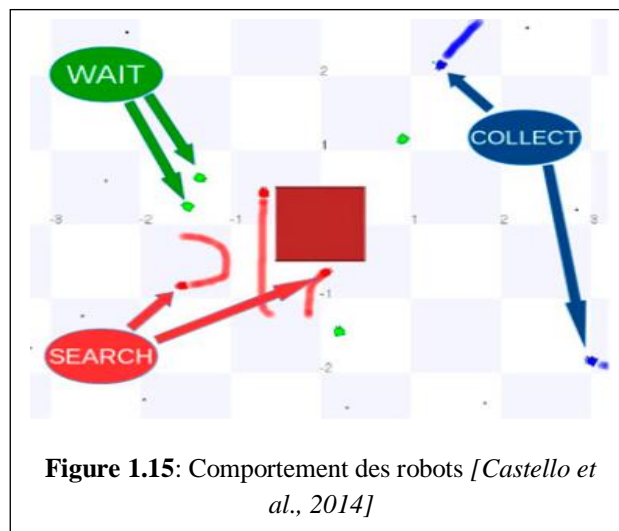
1. **Attente (wait)**: au début de la mission de foraging, tous les robots sont en état d'attente. Chaque robot initialisera une minuterie d'une durée fixée au début de la simulation. Si la durée expire, le robot détecte la valeur actuelle de stimulus (en fonction de quantité de nourriture  $F$  existante dans le dépôt,  $\text{Stimulus} = F(t) - F(t_0)$ ) et calcul sa probabilité de foraging ( $P_f$ ). Cette dernière, lui permet de décider de changer vers l'état *Rechercher (search)* ou bien de rester en attente.
2. **Rechercher (search)**: le robot commence par rechercher aléatoirement la nourriture dans son environnement. S'il localise un objet, il passe directement à l'état *Collecter (collect)*, sinon il continue sa recherche.
3. **Collecter (collect)**: une fois le robot détient un objet, il rentre au dépôt. Si le dépôt est atteint, il dépose l'objet et passe à l'état *Attente*.

Les simulations numériques montrent que le modèle proposé diminue considérablement l'écart moyen de nourriture à l'arène de l'essaim dans le cadre d'une mission de foraging multi-robots.

3) *Foraging optimization in swarm robotic systems based on an adaptive response threshold model [Castello et al., 2014]*

Castello et al. [Castello et al.,2014], ont proposé une extension du modèle ARTM (Adaptive Response Threshold Model) basé sur le DASM [Castello et al.,2013] pour l'allocation des tâches dans la mission de foraging. Le nouveau modèle ARTM est obtenu par la combinaison du FRTM (Fixed Response Threshold Model) [Yang et al.,2009] et le DASM [Kashiwagi et al.,2006], pour augmenter la capacité de survie du système.

Le but des robots est de satisfaire les demandes d'un système externe qui extrait la nourriture depuis les dépôts. Du fait que le travail est une extension du modèle propose dans [Castello et al,2013], la configuration reste la même ainsi que le comportement des robots dans le cycle de la mission de foraging (voir *Figure 1.15*).



Les résultats de simulation montrent que le modèle proposé augmente la survie du système pendant la mission de foraging.

#### **4) Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach [Castello et al., 2016]**

Ici, les auteurs proposent un algorithme de division des tâches entre les robots chargés de rechercher de la nourriture dans leur environnement. L'objectif est de maintenir une quantité de nourriture fixe et prédéfinie au préalable au niveau du dépôt centralisé qui servent à satisfaire les demandes d'un système externe.

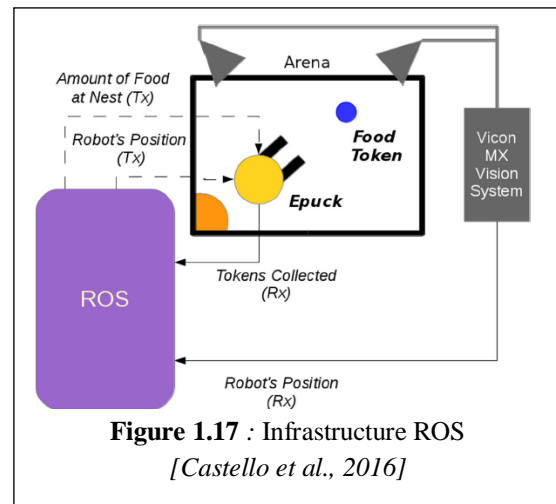
Ce travail vise à confirmer les avantages de l'utilisation de modèle de seuil dynamiques ARTM (capacité de survie,...), présenté dans [Castello et al., 2013][Castello et al.,2014] et tester par des simulations numériques seulement. Cette confirmation est faite par des expérimentations avec des robots réels.

Les auteurs proposent une version simplifiée de ARTM (SRTM Simple Response Threshold Model) pour améliorer les capacités d'adaptation et d'émergence de l'essaim de robots.

Autour de dépôt, il y a 5 robots de type *e-puck* et 10 objets sont distribués aléatoirement dans l'environnement (voir *Figure 1.16*), les robots sont commandés à l'aide du système ROS (Robot Operating System) (voir *Figure 1.17*), l'environnement est considéré comme un rectangle de surface  $2.0 \times 1.5$  mètres carré.



**Figure 1.16** : L'environnement considéré dans [Castello et al., 2016], avec dépôt : cercle jaune,



**Figure 1.17** : Infrastructure ROS [Castello et al., 2016]

Comme ce travail est une confirmation des résultats obtenus dans [Castello et al., 2013] [Castello et al., 2014], le comportement des robots reste le même (*wait, search, collect*) mais les auteurs ont considéré ici les problèmes des robots réels comme la collision entre robots. Les expérimentations réelles et de simulation numériques confirment que l'algorithme proposé est capable d'adapter le nombre de robots actifs, augmentant le taux de survie du système dans des missions de foraging, ainsi que réduire le temps de collision entre robots.

### 3.6. Le foraging multi-dépôts: travaux reliés

Nous présentons dans cette section les travaux récents et pertinents du foraging multi-dépôts.

#### 1) *Static Multiple Place Foraging Algorithm (MPFA<sub>static</sub>)* [lu et al., 2016]

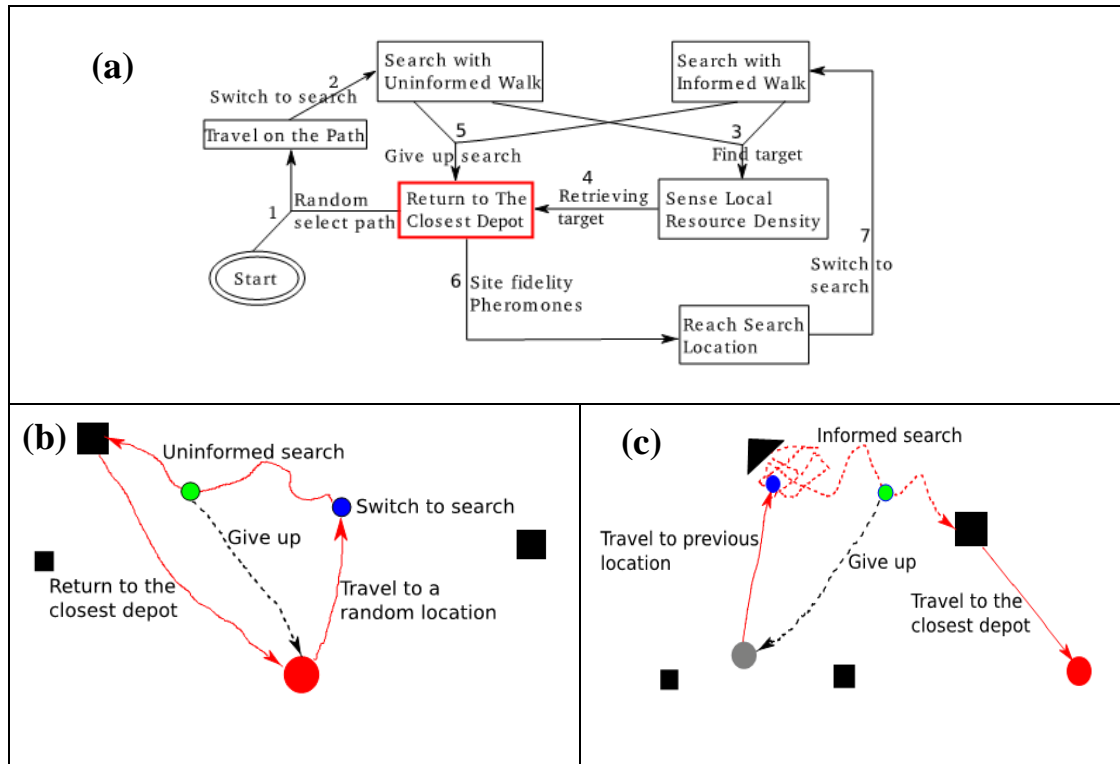
Lu et al. [lu et al., 2016] ont proposé un algorithme MPF nommé *Multiple Place Foraging Algorithm (MPFA)<sub>static</sub>*. Ce travail utilise plusieurs dépôts à partir desquels les robots partent et y retournent. Les auteurs utilisent un algorithme génétique (GA) pour faire évoluer les comportements de recherche collective des objets dans le simulateur

Autonomous Robots Go Swarming (ARGoS). Les robots utilisent tous le même Controller. Les auteurs utilisent des configurations environnementales de 2, 4 et 8 dépôts. Puis, ils testent la rapidité avec laquelle les ressources sont collectées à partir de trois distributions de ressources : aléatoires, partiellement et entièrement clustérisées. Ils observent par la suite l'effet du nombre de dépôts sur les performances, particulièrement : le temps de collisions, de recherche et de voyage (collision time, search time, travel time respectivement).

Le comportement d'un robot dans un MPFA<sub>static</sub> est donné par la *Figure 1.18(a)*. Le robot transit d'un état à l'autre lors de son foraging en imitant le comportement des fourmis naturelles. Cet algorithme forme une extension de l'algorithme CPFA [Hecker and Moses, 2015] avec plusieurs dépôts dispersés dans l'environnement. Dans MPFA<sub>static</sub> les robots sont répartis uniformément autour des dépôts. Ils démarrent d'un dépôt aléatoire, mais reviennent au dépôt le plus proche de leur position après avoir trouvé une ressource. Les robots ont à priori une connaissance de l'emplacement des dépôts. L'utilisation de plusieurs dépôts constitue la différence fondamentale entre le CPFA [Hecker and Moses, 2015] et le MPFA; tous les autres composants des deux algorithmes de foraging sont maintenus identiques afin de tester l'effet de l'utilisation de plusieurs dépôts sur l'efficacité du foraging. Comme dans la CPFA, les robots utilisent la *fidélité au site* ou suivent les *phéromones* pour exploiter les zones riches en ressources. Les robots partagent des points-de-passage de phéromones localement à leur dépôt actuel.

### 2) *Dynamic Multiple Place Foraging Algorithm (MPFA<sub>dynamic</sub>)* [lu et al., 2018]

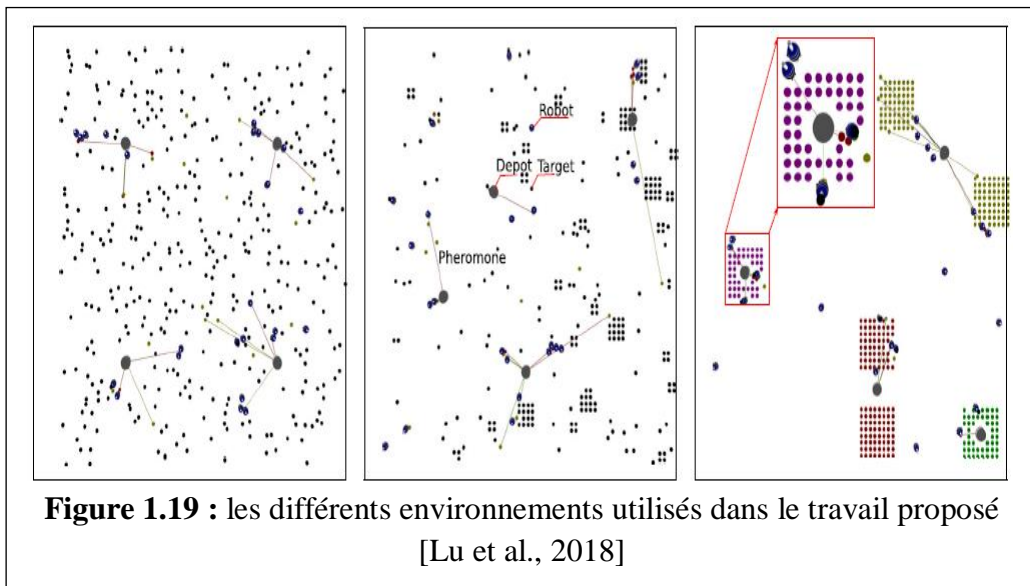
Lu et al. [lu et al., 2018], ont proposé un algorithme MPF nommé *Multiple Place Foraging Algorithm (MPFA)<sub>dynamic</sub>* avec des dépôts dynamiques, ces dépôts sont des robots spéciaux capable de transporter plusieurs objets. Les robots partent d'un dépôt pour chercher des cibles, puis reviennent au dépôt le plus proche pour livrer ces cibles (le dépôt le plus proche peut être différent de celui d'où le robot est parti). Les dépôts se déplacent vers de nouveaux emplacements en fonction de la position des cibles détectées par les robots. Les auteurs utilisent un algorithme génétique (GA) pour faire évoluer les comportements de recherche collective des objets dans le simulateur Autonomous Robots Go Swarming (ARGoS) dans plusieurs environnements qui ont des configurations différentes (voir figure 1.19).



**Figure 1.18 :** (a) Comportement des robots MPFA, (b) Un seul cycle de recherche sans information préliminaires et (c) Un seul cycle de recherche avec information préliminaires.

Dans ce travail, les auteurs comparent les performances du  $MPFA_{dynamic}$  avec l'algorithme  $MPFA_{static}$  proposé dans [lu et al., 2016] et le CPFA proposé dans [Hecker and Moses, 2015], 4 expériences ont été réalisées :

1. Dans la première expérience, comparaison du temps de recherche entre  $MPFA_{dynamic}$ ,  $MPFA_{static}$  et CPFA dans un environnement de 10\*10 mètres carré et contient 24 robots.
2. Dans la deuxième expérience, le test de scalabilité des algorithmes par l'augmentation de la surface de l'environnement 16 × 16 mètres carré avec 24 robots.
3. Dans la troisième expérience, le test de performance des algorithmes dans un environnement 50 × 50 mètres carré et 96 robots.
4. Dans la quatrième expérience, ils ont comptabilisé le transport effectué par les dépôts mobiles vers un seul point de collecte central. Dans cette expérience, chacun des quatre dépôts mobiles est un robot qui transporte les cibles vers un point de collecte central.



Nous avons comparé les travaux reliés, selon plusieurs critères que nous avons jugés utiles pour distinguer les travaux. Le *Tableau 1.1* montre la comparaison qualitative des travaux discutés précédemment.

Où :

- CPF* : Central Place Foraging (Foraging à dépôt central) ;
- MPF* : Multiple Place Foraging (Foraging à Multi-Dépôts) ;
- Durable* : Foraging durable (sustainable Foraging)
- $N_v$  : le nombre moyen de travail pour chaque robot
- $V_f$  : écart moyen de la nourriture dans le dépôt
- $Z_{rate}$  : le taux d'alimentation moyen des robots

Référence	CPF	MPF	Dépôts	Nombre de robots	Communication entre robots	Distribution des Objets	Durable	Stratégie d'exploration	Stratégie de Homing	Basé Intelligence en essaim	Mesures de performance	Plateforme de simulation	App réel
[Lu et al, 2016a]	Non	Oui	2, 4, 8	24	Indirecte (phéromones)	Aléatoire, partiellement, entièrement groupés.	Non	Marche aléatoire uniformément (ou <b>informée</b> ) corrélée	Suivre la lumière émise par le dépôt	Oui (Fourmis)	Temps de collision, de recherche et de voyage	ARGoS	Oui
[Castelo et al.,2013 ]	Oui	Non	1	10	-	Aléatoire	Oui	Marche aléatoire	coordonnées globales	Non	$N_v, V_f, Z_{rate}$	Libraire Stage	Non
[Castello et al.,2014]	Oui	Non	1	10	-	Aléatoire	Oui	Marche aléatoire	coordonnées globales	Non	Survie de système, $N_v, V_f, Z_{rate}$	Libraire Stage	Non
[Castello et al.,2016]	Oui	Non	1	5	-	Aléatoire	Oui	Camera intégré	coordonnées globales	Non	Survie de système, $N_v, V_f$	Libraire Stage	Oui
[Hecker& Moses,2015]	Oui	Non	1	6 physique 768 simulées	Indirecte (phéromones)	Aléatoire, partiellement, entièrement groupés	Non	Marche aléatoire uniformément (ou <b>informée</b> ) corrélée	Suivre la lumière émise par le dépôt	Oui (Fourmis)	Tolérance d'erreur, flexibilité, scalabilité	iAnt robot	Oui
[Lu et al.,2018]	Non	Oui	4	24,24,28,96	Indirecte (phéromones)	Aléatoire, partiellement, entièrement groupés.	Non	Marche aléatoire uniformément (ou <b>informée</b> ) corrélée	Suivre la lumière émise par le dépôt	Oui (Fourmis)	Temps de collision, de recherche et de retour, et scalabilité	ARGoS	Non

**Tableau 1.1** : Comparaison qualitative des travaux reliés.



D'après le *Tableau 1.1*, on peut faire plusieurs constatations :

1. La majorité des travaux ont utilisé la marche aléatoire pour l'exploration ;
2. Les travaux du CPF ou MPF ont considéré le foraging classique.
3. 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 ;
4. Quelques travaux utilisent les algorithmes bio-inspirés pour améliorer l'exploration et imiter le comportement de communication indirecte non couteux des insectes sociaux;
5. Les critères de performances changent du travail à l'autre, mais le critère le plus important dans les travaux qui ont durable est la survie du système.

L'objectif premier de la nouvelle extension du foraging classique qui est le foraging durable est de proposer des systèmes de foraging qui rapproche le maximum possible des applications du monde réel. La comparaison des travaux dans le *Tableau 1.1* montre déjà que ce problème a été seulement considéré par les travaux de Castello dans le cadre du CPF. Castello, n'as pas considéré l'intelligence en essaim dans ces travaux, et notre travail ici c'est de revoir le problème de foraging durable avec une vision intelligence en essaim dans laquelle on va utiliser un algorithme d'interaction inspiré du comportement des lucioles.

#### 4. Conclusion

Ce chapitre a été divisé en quatre parties. Dans la première, nous avons présenté quelques définitions sur le domaine de l'intelligence en essaim. Dans la deuxième partie, nous avons défini le domaine de la robotique en essaim ainsi que quelques comportements de base qui en découle. Dans la troisième partie, nous avons présenté le problème du foraging avec présentation d'une taxonomie qui montre les différents types de foraging existants dans la littérature. La quatrième partie a été dédiée aux travaux reliés CPF et MPF qui considèrent ou non le problème de durabilité du nid. Nous avons terminé cette partie par un tableau de comparaison qui nous a permis de faire plusieurs constatations importantes pour proposer une solution adéquate.

# Chapitre 02 : Conception

## 1. Introduction

Dans un CPF ou un MPF classique, plusieurs robots sont chargés de chercher, collecter et transporter un ou plusieurs objets à un ou plusieurs dépôts. Les robots adoptent une stratégie de marche aléatoire ou stratégique pour localiser les objets. Dans cette définition classique du foraging, les robots cherchent à maximiser le nombre d'objets collectés, de réduire le temps global de foraging, de minimiser l'énergie consommée ou encore de réduire la distance parcourue. Dans l'objectif de rapprocher encore le problème de foraging du monde réel, Castello [Castello, 2016], a proposé une nouvelle extension du foraging classique nommée foraging durable (sustainable foraging). Dans un tel foraging, les objets stockés dans les dépôts, sont retirés à plusieurs reprises par des systèmes externes, et que le système de foraging doit maintenir sa stabilité (la survie de tous les dépôts) pour des longues périodes.

Dans ce travail, nous avons proposé d'utiliser un modèle d'interaction entre robots basé intelligence en essaim (Algorithme de Firefly). Ce modèle permet un recrutement de robots appartenant à des dépôts étrangers en cas de nécessité.

On commence ce chapitre par présenter la problématique de recherche qui nous intéresse dans ce travail. Par la suite, nous présentons l'objectif à atteindre ainsi qu'une présentation générale de la solution qu'on a proposée. Notre solution dépend de deux algorithmes bio-inspirés : lévy walk et Firefly, nous consacrons donc une section qui explique les fondements de ces deux algorithmes. Ensuite, nous focalisons sur l'algorithme proposé, où nous présentons la structure paramétrique, le comportement de nos robots à travers un organigramme, et les pseudos codes des différents comportements. On termine le chapitre par une conclusion.

## 2. Problématique

Bien que le foraging classique forme une abstraction de plusieurs applications du monde réel comme : le nettoyage, le déminage, l'exploration des environnements hostiles, la recherche et le sauvetage, son application sur monde réel reste complexe et loin d'être concrétisée.

Aussi, diverses caractéristiques sont nécessaires pour répondre aux exigences du monde réel tel que : un mécanisme d'adaptation en ligne, la durabilité du système et l'adaptabilité dans des scénarios critiques reste presque non étudié dans le domaine. Encore, le foraging multi-dépôts n'as pas été mené dans des scénarios où les robots doivent répondre aux exigences d'un système externe.

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 une collaboration entre robots appartenant à des dépôts étrangers.
2. Quel modèle d'interaction faut-il utiliser pour garantir une meilleure collaboration qui ne demande pas de communication couteuse ?

### 3. Objectifs et Propositions

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

1. Proposer un système de foraging multi-dépôts pour des systèmes critiques en temps réel.
2. Utiliser un modèle d'interaction qui permet une collaboration efficace entre les robots des dépôts étrangères ;

Nous avons donc :

3. 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 Lévy Walk pour l'exploration et la recherche des objets grâce à son efficacité à explorer la totalité de l'espace par rapport aux autres techniques aléatoires ;
4. Si un manque a été détecté dans l'un des dépôts par l'un de ses robots, ce dernier utilise une attraction lumineuse basée sur l'algorithme Firefly, pour 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. Analyse des Algorithmes reliés

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

### 4.1. Algorithme Lévy Walk

Le Lévy Walk (marche de Lévy), nommé au l'honneur du mathématicien français Paul Lévy, qui a découvert dans les années 1930 une classe de distribution de probabilité avec un moment infini. Lévy Walk est une distribution de recherche aléatoire dans laquelle les longueurs de pas sont définies par une distribution de probabilité. La distribution de probabilité de Lévy s'écrit sous la forme [Lévy, 1937][Gnedenko et al .,1954] :

$$P_{\alpha,\gamma}(x) = \frac{1}{\pi} \int_0^{+\infty} e^{-\gamma q^\alpha} \cos(qx) dq, x \in \mathbb{R} \quad (1)$$

Cette distribution est symétrique pour  $x = 0$ , avec  $\alpha$  est le facteur de control de la distribution compris entre 0 et 2, et  $\gamma$  est le facteur d'échelle.

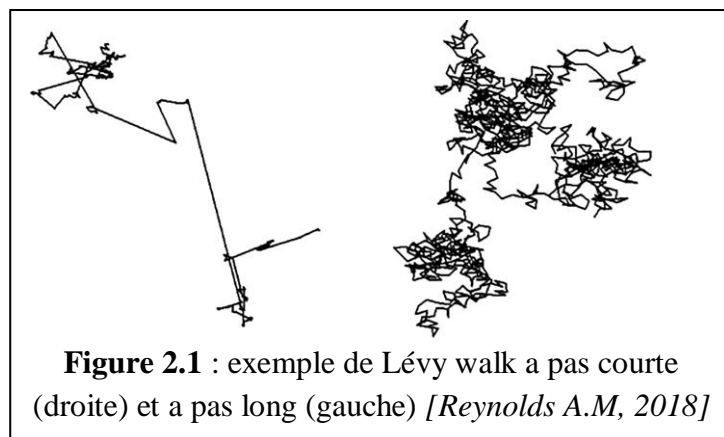
On fixe  $\gamma = 1$ , la distribution de probabilité peut être approximé par :

$$P_\alpha(x) \approx x^{-\alpha-1} \quad (2)$$

La longueur des pas  $l$  est générer par l'équation suivante :

$$l = l_0 \left( \frac{1}{\beta^\alpha} - 1 \right) \quad (3)$$

Où,  $l_0$  est la longueur initiale et  $\beta \in [0,1]$ .



### 4.2. Algorithme Firefly

Les lucioles (firefly), sont des petits coléoptères ailés capable de produire une lumière clignotante à l'intérieur de leur corps pour une attraction mutuelle. Dans un essaim, la luciole peut être influencée par d'autres lucioles.



L'algorithme des lucioles (FireFly Algorithm) est une Méta-heuristique récente développée par XS Yang [Yang ,2009] et inspiré par l'atténuation de la lumière sur la distance et l'attraction mutuelle.

Les règles idéalisées de l'algorithme de Lucioles sont [Yang ,2009] :

1. Toutes les lucioles sont unisexes.
2. L'attractivité des lucioles est proportionnelle à la luminosité.
3. La luminosité de la lumière clignotante peut être considérée comme une fonction objective qui devra être optimisé.

Dans un milieu donné, l'intensité lumineuse est définie par l'équation :

$$I = I_0 e^{-\gamma r} \quad (4)$$

Où :  $\gamma$  le coefficient d'absorption,  $r$  la distance.

L'attractivité d'une luciole varie selon l'intensité de la lumière des lucioles adjacentes, elle est définie par :

$$\beta = \beta_0 e^{-\gamma r^2} \quad (5)$$

Où :  $\beta_0$  est l'attractivité quand  $r = 0$ .

La distance entre deux lucioles  $i$  et  $j$  est donnée par la norme euclidienne entre les deux vecteurs qui contiennent la position de chaque luciole

1. La distance entre deux lucioles  $i$  et  $j$  est défini par :

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (6)$$

2. Le mouvement d'une luciole  $i$  vers une autre luciole  $j$  est défini par l'équation :

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha (\text{rand} - \frac{1}{2}) \quad (7)$$

## 5. Algorithme proposé : Lévy Walk Firefly Recruiting Algorithm

Nous présentons dans cette section, l'algorithme de foraging que nous avons proposé : **Lévy Walk Firefly Recruiting Algorithm (LWFRA)**. Nous commençons par montrer la structure paramétrique de LWFRA, ensuite nous présentons une description détaillée de comportement des robots LWFRA et nous terminerons la section par les pseudos codes associés.

### 5.1. Structure paramétrique de LWFRA

Dans l'algorithme LWFRA on a besoin de gérer : la recherche des objets, la gestion de la lumière, le mouvement d'un robot vers un autre, la vitesse de déplacement, et la distance de communication.

1. Pour la recherche des objets dans l'environnement, nous avons utilisé l'algorithme LW pour générer des marches aléatoires. Pour cela, les pas aléatoires sont générés selon l'équation (8) :

$$l = l_0 \left( \frac{1}{\beta^\alpha} - 1 \right) \quad (8)$$

2. Pour la gestion de la lumière, nous avons utilisé les règles suivantes :

- Augmenter l'intensité lumineuse à chaque fois l'état *recrutement* est déclenché. L'intensité sera en fonction du nombre des objets manquants dans le dépôt et qui doivent être récupérer pour garantir la survie du dépôt. L'intensité est définie par l'équation (9) :

$$I = F(O_m) \quad (9)$$

- L'intensité est remise à 0 si le cycle de recrutement est terminé  $I = 0$ .
- Réduire l'intensité à chaque fois un robot est recruté selon l'équation (10) :

$$I_{t+1} = I_t - 1 \quad (10)$$

3. Le déplacement d'un robot vers un autre se fait selon l'équation (11) :

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha' (rand - \frac{1}{2}) \quad (11)$$

4. La vitesse de déplacement d'un robot augmente lorsqu'une lumière est détectée selon l'équation (12) :

$$v_{t+1} = \tau * v_t \quad (12)$$

5. La distance de communication entre deux robots est calculée selon la formule (13) :

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (13)$$

Le *Tableau 2.1* résume les abréviations utilisées dans cette partie et leur signification :

<b>Abréviation</b>	<b>Définition</b>
$l_0$	Longueur de pas initial
$l$	Longueur de pas calculée
$\beta$	Facteur d'échelle
$\alpha$	Nombre aléatoire appartient à l'intervalle [0 ;1]
$\beta_0$	Facteur d'attraction
$I$	Intensité lumineuse
$O_m$	Nombre des objets manquant
$x_i$	Position de robot i
$\gamma$	Facteur d'échelle
$r_{ij}$	Distance entre deux robots i et j
$\alpha'$	Coefficient de poids
$rand$	Un nombre aléatoire (random)
$\tau$	Coefficient de poids

*Tableau 2.1* : Abréviations utilisées dans l'algorithme LWFRA

## 5.2. Comportement des robots LWFRA

Dans cette partie, nous présentons tout d'abord une description textuelle des différents états. Par la suite, nous présentons l'organigramme représentant le comportement des robots LWFRA pendant la mission de foraging. Nous terminerons par les pseudos-codes associés.

### 5.2.1. Description textuelle des différents états

- **Démarrer** : Au début de la mission de foraging, tous les robots sont distribués autour des dépôts dans l'environnement avec une intensité lumineuse égale à zéro, les dépôts sont positionnés d'une manière équidistante, chaque dépôt au démarrage contient la même quantité d'objets noté par  $F(t_0)$ , cette quantité commence à diminuer en fonction du temps, ce qui simule l'extraction des objets par un système externe. Les objets à collecter sont distribués d'une manière aléatoire ou bien par des regroupements (cluster) dans l'environnement.
- **Attente** : dans cet état, le robot reste dans son dépôt original, en attente d'être activé pour le foraging. A chaque instant la décision de changer vers l'état *chercher* est basée sur le calcul de fonction de décision, cette dernière est calculée en fonction de quantité d'objets extraite par le système externe, on note par  $S(t)$  le stimulus qui est la différence entre la quantité d'objets dans le dépôt à l'instant  $t_0$  et à l'instant  $t$ , il est défini par :

$$S(t) = F(t) - F(t_0) \quad (14)$$

A partir de cette formule, on peut définir la fonction de décision  $F_d$  par :

$$F_d = \begin{cases} \textit{Attente} & \text{si } S(t) < 0 \\ \textit{chercher} & \text{si } \varepsilon * F(t_0) < S(t) < \theta * F(t_0) \\ \textit{recruter} & \text{si } S(t) > 0.4 * F(t_0) \end{cases} \quad (15)$$

- **Chercher** : dans cet état, le robot commence à parcourir l'environnement avec une marche LW. Chaque robot est capable de transporter un objet à la fois. Pendant cet état le robot peut entrer en communication avec un autre robot, si l'intensité de ce dernier est supérieure à zéro et qu'il appartient à un autre dépôt, donc le robot déplace envers ce dernier par l'équation de mouvement (*équation 10*) pour recevoir les coordonnées de nouveau dépôt. Le robot est donc recruté pour transporter les objets vers le nouveau dépôt afin de le sauver et de garantir sa survie et la survie du système ainsi. Si aucun recrutement n'a été fait, le robot collecte les objets et les transporte à son dépôt original.



- **Collecter** : Dans cet état, le robot collecte l'objet trouvé et revient à son dépôt où il le dépose, puis il passe à l'état attente et calcule la fonction de décision (selon l'équation 15). Si le robot est recruté, il dépose l'objet et supprime les coordonnées du dépôt recruteur, puis il passe à l'état chercher pour son dépôt original.
- **Recruter** : Dans cette état, l'intensité de robot devient égale à la quantité des objets manquants (équation 9). Le robot commence à chercher des objets dans son environnement ainsi que des robots pour les recruter, on a proposé de recruter  $O_m$  robots, si le robot rencontre un robot dans son voisinage qui a une intensité nulle et appartient à un autre dépôt, donc le robot le recrute en envoyant les coordonnées de son dépôt et continuer à chercher d'autre robots jusqu'à ce que l'intensité deviennent nul. Si l'état de recrutement est terminé, le robot commence à chercher des objets pour le déposer dans son dépôt.

### 5.2.2. Organigramme de LWFRA :

La figure 2 .3 ci-dessous représente le comportement des robots LWFRA.

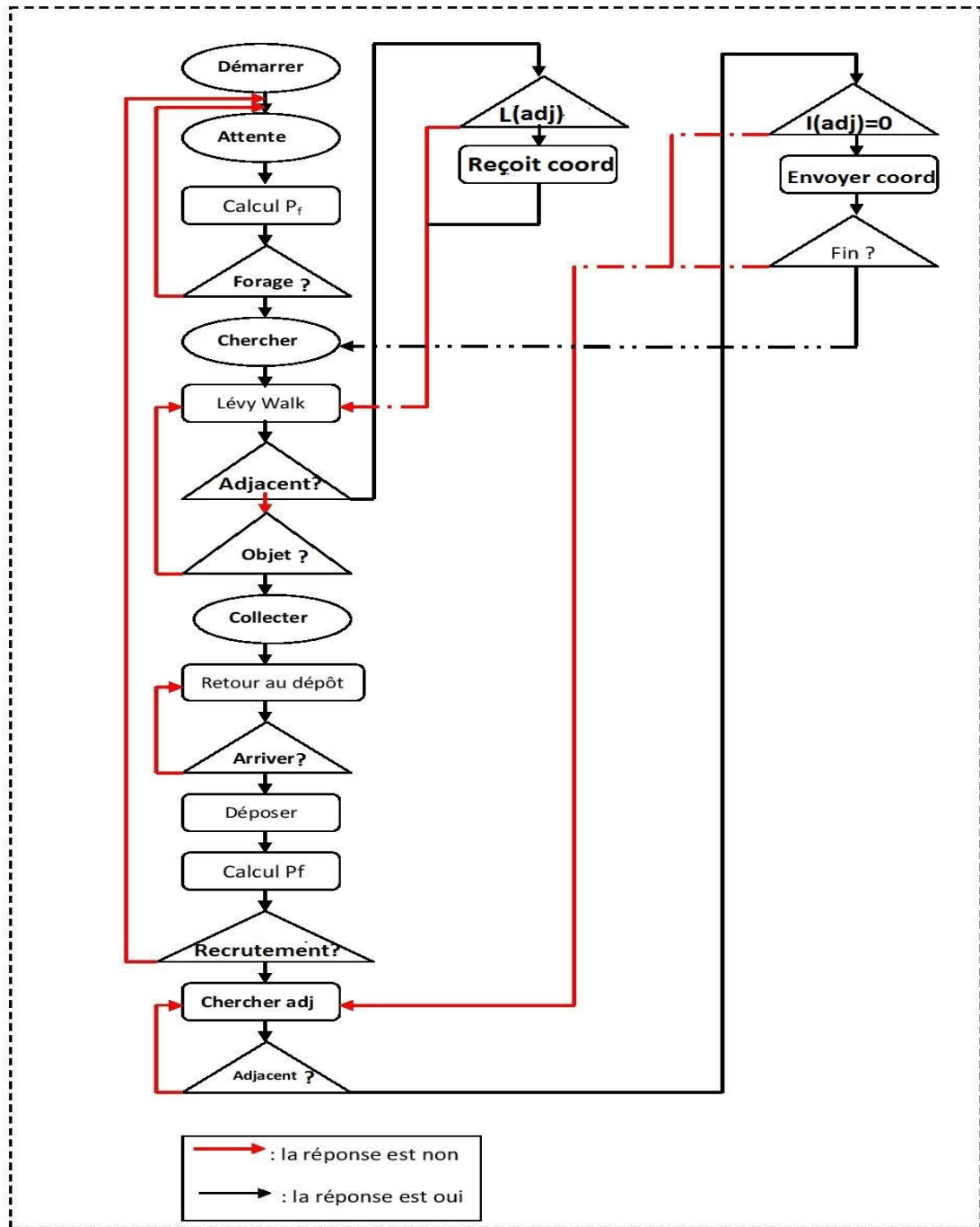


Figure 2.3 : Organigramme représentant le comportement des robots LWFR

### 5.2.3. Pseudo code de l'algorithme LWFR

Comme il est montré dans l'organigramme de la Figure 2.3, l'algorithme LWFR se compose de quatre états principaux : Attente, chercher, collecter, recruter. Les algorithmes 2.1, 2.2, 2.3, 2.4 montrent respectivement le pseudo code de chacun des états mentionnés.

### Algorithme 2.1 : Robot en attente

```
1 : Tant que ( $F_d(t) = Attente$ ) Faire
2 : Rester autour de dépôt
3 :  $t = t + 1$ 
4 : Fin Tant que
5 : Si  $F_d(t) = chercher$  Alors Aller vers Chercher (Algorithme 2.2)
6 : Fin si.
```

### Algorithme 2.2 : Chercher

```
1 : Tant que ( $\nexists robot\_adj$  et  $\nexists objet$ ) Faire
2 : Marche utilisant LW (équation (8))
3 : Fin Tant que
4 : Si ( $\exists objet$ ) Alors Aller vers Collecter (Algorithme 2.3)
5 : Sinon Si ( $\exists robot\_adj$ ) Alors Début
6 :           Si Couleur (robot_adj) = rouge et robot_adj  $\in$  autre dépôt Alors
7 :           Aller vers robot_adj utilisant équation (11)
8 :           Recevoir (cordonnées)
9 :           Sinon Continuer à Chercher.
10 :          FinSi
11 :          Fin
12 : FinSi
```

### Algorithme 2.3 : Collecter

```
1 : Prendre objet
2 : Tant que ( $\nexists arrive$  au dépôt)
3 : continue a marché vers le dépôt adéquat
4 : Fin Tant que
5 : Déposer objet
6 : Si le robot est recruter Alors Début
7 :           attribuer le robot à leur dépôt original
8 :           Aller vers Chercher (Algorithme 2.2)
9 :           Fin
10 : Sinon Début
```

```
11 :      calcul  $F_d(t)$ 
12 :      Si  $F_d(t)$  = Attente Alors Aller vers Robot en attente (Algorithme 1)
13 :      Sinon Si  $F_d(t)$  =chercher Alors vers Chercher (Algorithme 2)
14 :      Sinon Aller ver Recruter (Algorithme 2.4)
15 :      FinSi
16 :      Fin
17 : Fin si.
```

### Algorithme 2.4 : Recruter

```
1 : Tant que nombre_robot_recruter <  $O_m$  faire
2 : Tant que ( $\exists$  robot_adj) Faire
3 : continuer à chercher un robot_adj
4 : Fin Tant que
5 : Si Couleur (robot_adj) <> rouge et robot_adj  $\in$  autre dépôt Alors
6 :   Début
7 :   Envoyer(cordonnées)
8 :   diminuer l'intensité (équation (10))
9 :   Fin
10 : FinSi
11 : Fin Tant que
12 : Aller vers Chercher (Algorithme 2.2)
```

On a considéré un environnement sans obstacles. Le seul obstacle existant c'est les frontières de l'environnement. Si le robot arrive sur une frontière ou rencontre un autre robot dans son chemin, il applique une rotation et change de direction pour les éviter.

## 6. Conclusion

Dans l'objectif de rendre les systèmes de foraging plus proche du monde réel par l'intégration de certaines caractéristiques importantes comme : l'adaptation en ligne, la durabilité du système et l'adaptabilité dans des scénarios critiques, et vu que l'utilisation de la durabilité dans un système de foraging multi-dépôt a été réalisée par un seul travail jusqu'à aujourd'hui, nous avons eu le courage d'investiguer cette nouvelle question, voir direction de recherche.

Nous avons donc proposé un algorithme de foraging multi-dépôts qui réponds à des exigences d'un système externe. L'algorithme proposé utilise le Lévy Walk comme stratégie d'exploration, mais aussi l'algorithme firefly comme modèle d'interaction pour attirer autant de robots vers le dépôt en alerte. L'hybridation de LW et FF faite dans ce travail tente

d'augmenter la survivabilité des dépôts dans un système de foraging multi-dépôts. Un modèle d'interaction basé intelligence en essaim était en effet, la solution pour une meilleure et non couteuse collaboration entre les robots des dépôts étrangers.

Nous avons présenté dans ce chapitre les deux algorithmes reliés : Lévy Walk et Firefly. Puis, l'algorithme proposé LWFRA, son organigramme, une description de ses états et les pseudos codes associés.

Dans le chapitre suivant, nous passons à une implémentation de cette proposition ainsi qu'une expérimentation quantitative de l'algorithme proposé.

## Chapitre 03 : Implémentation et Expérimentations

### 1. Introduction

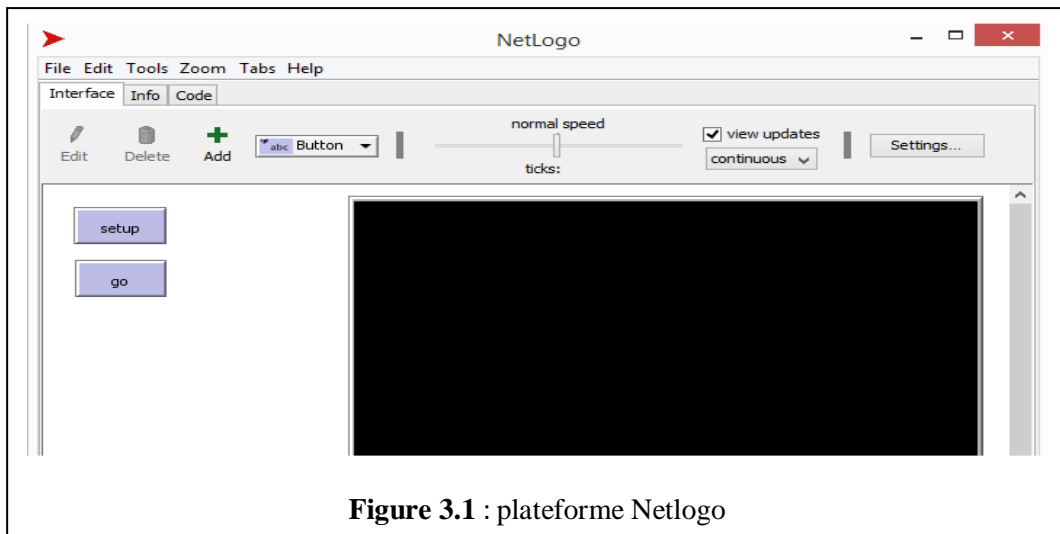
Ce chapitre a été consacré à l'implémentation, l'étude expérimentale, et les discussions des résultats obtenues par l'algorithme LWFR proposé dans le chapitre précédent. Pour des raisons de limite de temps, nous avons utilisé la plateforme de simulation Multi-Agents Netlogo.

Nous avons réalisé plusieurs simulations dans des configurations environnementales différentes, ainsi qu'une comparaison quantitative qui montre la supériorité de notre proposition.

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

### 2. Environnement de développement Netlogo

Les simulations réalisées ont été implémentées et exécutées sous la plate-forme Multi-Agents Netlogo. NetLogo est un environnement de modélisation programmable permettant de simuler des phénomènes naturels et sociaux. Il a été écrit par Uri Wilensky en 1999 [Wilensky U, 1999] et est en développement continu depuis au Centre pour l'apprentissage connecté et la modélisation par ordinateur. Il est particulièrement bien adapté à la modélisation de systèmes complexes évoluant dans le temps. Les modélisateurs peuvent donner des instructions à des centaines, voire des milliers d'agents, tous indépendants. Cela permet d'explorer le lien entre le comportement au niveau micro des individus et les schémas au niveau macro qui émergent de leur interaction.



Dans la Figure 3.1, on peut voir trois onglets principaux :

1. **Interface** : où la simulation est affichée et l'utilisateur final peut interagir à l'aide des curseurs, boutons, etc. disponibles.
2. **Info** : Où on peut trouver des informations générales sur le modèle. Il s'agit essentiellement de la page d'aide du modèle.
3. **Code** : Emplacement du code Netlogo pour un modèle

Une simulation NetLogo consiste en un monde constitué de rectangles (en 2D) ou de blocs (en 3D) appelés «patches» au sein desquels des agents mobiles appelés «turtles» peuvent se déplacer et évoluer en fonction des patches les entourant et de ceux rencontrés dans leur chemin.

Dans Netlogo, il existe trois types d'agents :

1. **Patch** : Un petit rectangle (en 2D) ou un bloc (en 3D) du monde de la simulation. Un patch est identifié par ses coordonnées,  $(x, y)$  ou  $(x, y, z)$ . Ainsi, il ne peut pas se déplacer, mais il peut contenir des variables (par exemple, sa couleur, depuis combien de temps il a été visité, ...).
2. **Turtle** : Un agent mobile qui peut passer d'un patch à un autre en fonction des règles définies pour son déplacement. Puisqu'il peut se déplacer, Les tortues peuvent également contenir des variables et il existe différents types de tortues (différents types d'agents de tortues, par exemple des lapins et des lièvres).
3. **Link** : Une connexion qui peut être créée entre deux agents de tortue. Il apparaît comme une ligne correspondant au chemin le plus court entre les deux tortues. Un lien

est identifié par le numéro de qui des deux tortues qu'il lie. Les liens peuvent contenir des variables.

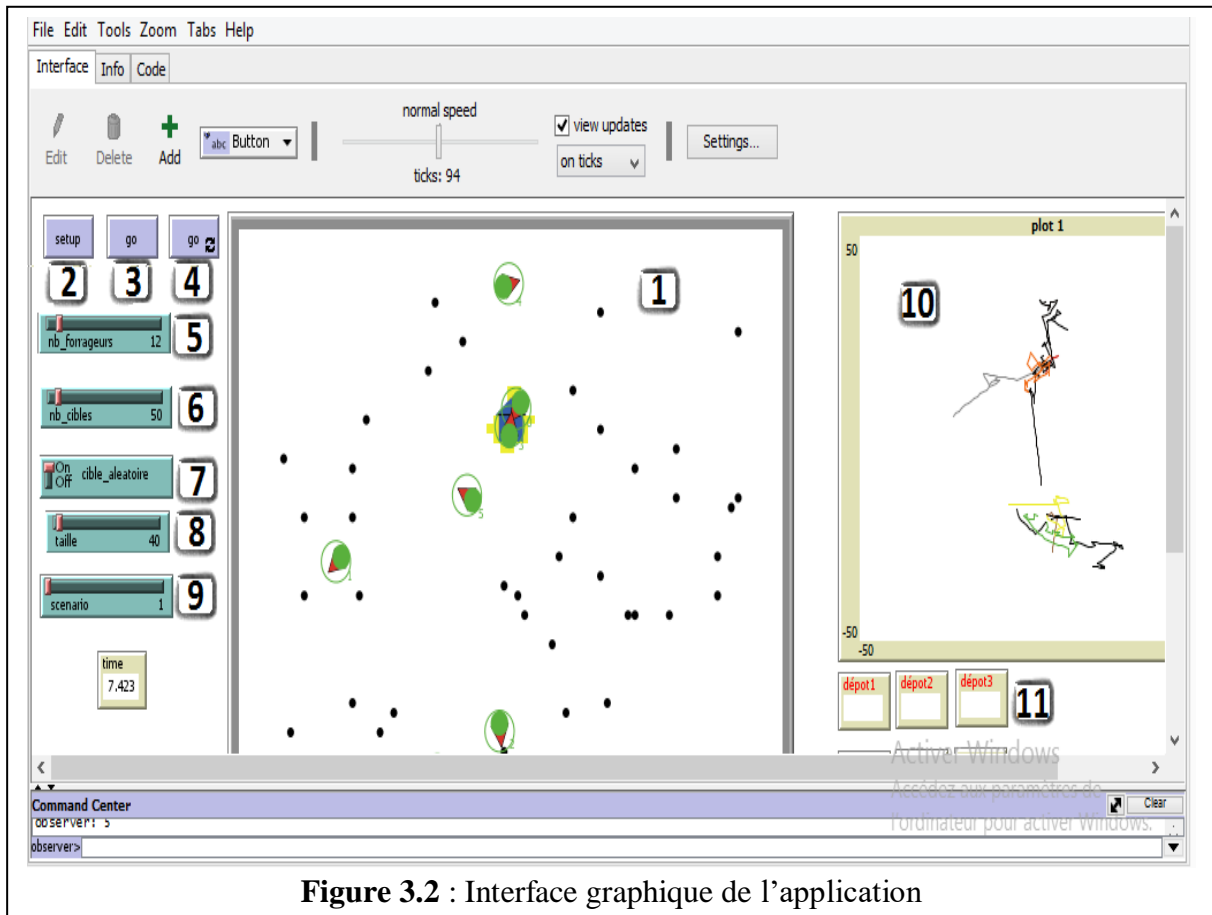
### 3. Présentation générale de l'application

#### 3.1. Interface graphique du système

La *Figure 3.2* montre l'interface principale de notre application. La configuration d'une simulation est sauvegardée sous forme d'un modèle, avec l'extension « *nlogo* ». Ce modèle contient tous les modules (fonctions et procédures) pour bien fonctionner les agents, et les paramètres spécifiques à la simulation tels que la durée de la simulation, la position des dépôts et des agents, le nombre de cibles distribuées, la taille de l'environnement...etc. Après le chargement et l'exécution du modèle *nlogo*, la fenêtre dans la *Figure 3.2* apparaît et contient les éléments suivants :

1. **Environnement de simulation** : c'est un espace limité, qui contient les agents, les cibles et les dépôts.
2. **Le bouton Setup** : pour initialiser le modèle, c'est-à-dire définir les conditions de départ de modèle, comme la taille de l'environnement, le nombres de objets....
3. **Le bouton Go** : pour lancer la simulation et exécuter les commandes une seule fois seulement.
4. **Le bouton Go Forever** : pour exécuter les commandes d'une façon répétitive jusqu'à ce que la condition d'arrêt de la simulation soit satisfaite.
5. **Le slider Nombre de fourrageurs** : permet à l'utilisateur de définir le nombre des agents dans la simulation.
6. **Le slider Nombre d'objets** : permet de définir le nombre d'objets dans l'environnement.
7. **Le commutateur Cibles aléatoire** : permet de définir la façon de la distribution des objets dans l'environnement, soit aléatoire, soit par regroupement (cluster).
8. **Le slider taille** : permet de définir la taille de l'environnement de simulation.
9. **Le slider scénario** : permet de définir quel scénario exécut, on a proposé 3 scénarios : scénario 1, scénario 2 et scénario 3 (avec 2 ,4 ,6 dépôts respectivement).
10. **Plot** : une représentation graphique de trace de déplacement de chaque agent dans l'environnement pendant la simulation.
11. **Les afficheurs Dépot1, Dépôts2, ...** : pour afficher la quantité des cibles dans chaque dépôt.





**Figure 3.2 :** Interface graphique de l'application

### 3.2. Les composants de notre système et leur Caractéristiques

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

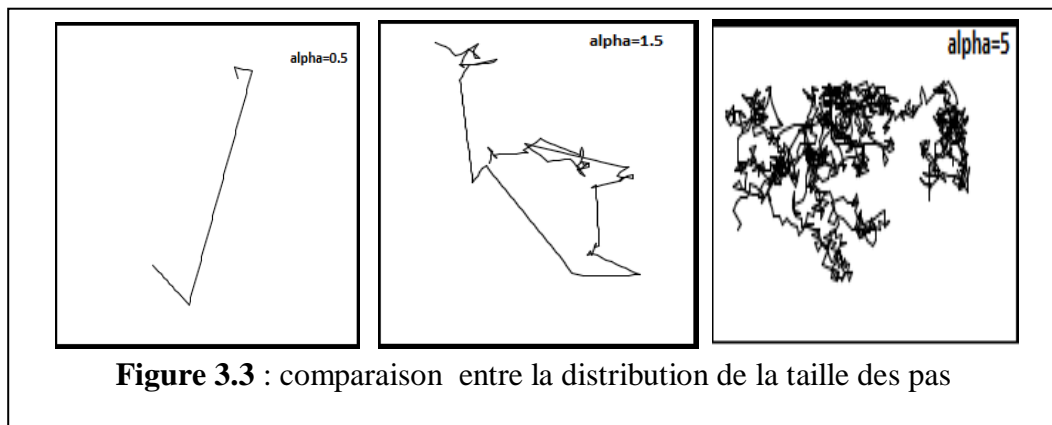
- **L'environnement** : est modélisé par un espace 2D continu limité par bordure.
- **Les robots** : sont modélisés par des agents d'une forme de type « Z\_H ». L'agent utilisé dans nos simulations dispose des caractéristiques suivantes : une taille de 4 patches, une capacité de collection égale à 1, une vitesse de déplacement égale à *un pas par tick*, une distance de vision égale à 15 patches, une distance de communication avec les autres agents, égale à 4 patches. Dans le cas recrutement, l'agent change sa couleur à la couleur *rouge* et sa taille devient 5 patches. Dans le cas de déplacement vers un autre agent pour recevoir des coordonnées, sa couleur devient *jaune*.
- **Les objets** : sont représentés par des cercles noirs. Ils sont de taille égale à 1 patch. Les objets sont régénérés et distribués dans l'environnement chaque 1 minute.

- **Les dépôts** : sont représentés par des formes de type « House ». Ils sont de taille 5 *patches*, initialisés à une quantité d'objets qui égale à 20 *objets*. La quantité de chaque dépôt diminue après une période de temps (la durée est fixée pour chaque dépôt dans les scénarios de simulation).
- La lumière est représentée par la couleur rouge d'agent.

### 4. Etude de paramètres de l'algorithme LWFR

Pour assurer des résultats efficaces, il faut une bonne initialisation de certains paramètres des deux algorithmes. Dans cette section, on va discuter les valeurs de quelques paramètres de notre algorithme LWFR tel que la longueur de pas.

- Commençons la longueur du pas dans l'algorithme LW. La *Figure 3.3* montre une comparaison entre la distribution de la taille des pas.



D'après cette *Figure 3.3*, le meilleur choix est  $\alpha = 0.5$ , pour assurer un bon parcours ou bien une couverture plus large de l'environnement par le robot.

- Pour assurer le déplacement d'un robot vers un autre, on a utilisé une fonction prédéfinie dans Netlogo s'appelle « FACE », donc aucun paramètre a étudié dans ce cas.
- Dans la simulation, on a initialisé chaque dépôt par une quantité égale à 20 *objets*, dans l'équation (15), il y a deux facteurs de décision, en fonction de ces facteurs le robot décide d'aller *chercher seulement* ou bien *recruter* un autre robot. Comme la valeur initiale de dépôt est égale à 20 *objets*, notre choix été de prendre la valeur de  $\varepsilon = 0.2$  et  $\theta = 0.4$ .

## 5. Simulation et analyse des résultats

### 5.1. Critères de performances

Nous avons utilisé les deux critères de performances suivant :

1. *Le taux de survie de dépôt* : est défini par l'équation (16):

$$taux_{(\phi)} = \sum_{i=1}^N \frac{S(\phi, i)}{N} \quad (16)$$

$$S(\phi, i) = \begin{cases} 1 & \text{si } F(\phi, i) \geq 0 \\ 0 & \text{si } F(\phi, i) < 0 \end{cases} \quad (17)$$

*Sachant que N est le nombre total d'expérimentations.*

2. *La survie globale de système* : est défini par l'équation (18) :

$$S_{globale} = \sum_1^{\omega} \frac{taux_{(\phi)}}{\omega} \quad (18)$$

*Où,  $\omega$  représente le nombre total des dépôts utilisés.*

### 5.2. Scénarios de simulation

Pour tester les performances de LWFR, on a réalisé trois scénarios de simulation, nous avons résumé ces scénarios dans le *Tableau 3.1*.

<b>Scénario 1 : variations de nombre de dépôts</b>
Taille de l'environnement : $50 p \times 50p$ Nombre de robots : 5 par dépôt Nombre de cibles : 50 régénérer Nombre de dépôts : 2, 4, 6
<b>Scénario 2 : variations de nombre de robots</b>
Taille de l'environnement : $50 p \times 50p$ Nombre de cibles : 50 régénérer Nombre de dépôts : 4 Nombre de robots : 5, 10, 20,30
<b>Scénario 3 : variation de taille de l'environnement</b>
Taille de l'environnement : $30 p \times 30p, 50p \times 50p, 80p \times 80p, 100p \times 100p$ Nombre de robot : 5 par dépôt Nombre de cibles : 50 régénérer Nombre de dépôts : 4

**Tableau 3.1** : les trois scénarios de simulation.

1. Le scénario 1 permet de tester les performances en termes de survie globale quand nous varions le nombre de dépôts 2, 4 et 6.
2. Le scénario 2 permet de tester les performances en termes de survie de chaque dépôt et la survie globale quand nous varions le nombre de robots de 5 à 20 ;
3. Le scénario 3 permet de tester les performances en termes de survie de chaque dépôt et la survie globale quand nous varions la taille de l'environnement de  $30 p \times 30 p$  à  $100 p \times 100 p$ .

### 5.3. Résultats, discussions et comparaisons

Nous avons comparé l'algorithme LWFR a pas court avec LWFR a pas long, nous présentons dans ce qui suit les résultats obtenus dans les différents scenarios, nous discutons et analysons les résultats obtenus.

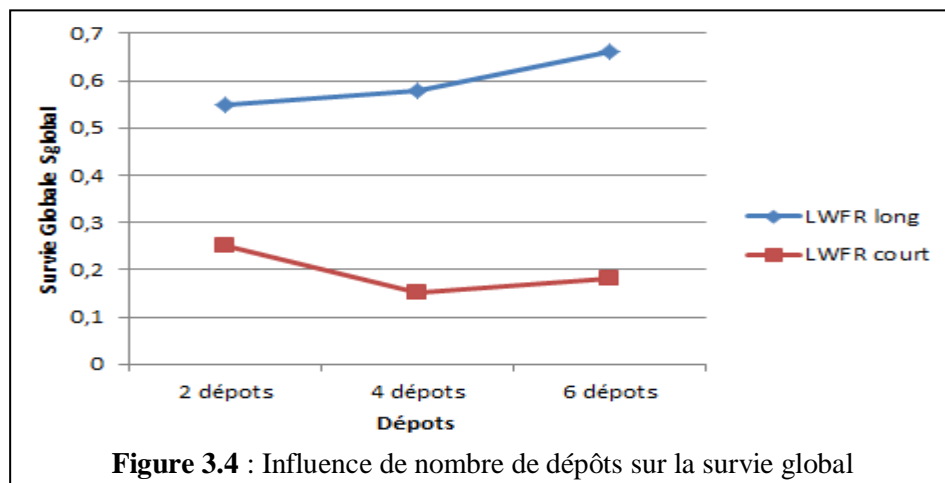
#### 5.3.1. Scénario 1 :

Dans ce scénario, nous avons étudié l'influence de la variation du nombre de dépôts sur les performances de LWFR a pas court et LWFR a pas long. Nous avons utilisé 2, 4, 6 dépôts.

Les différents paramètres de ce scénario sont détaillés dans le *Tableau 3.1*.

Dépôts \ Algorithme	2 dépôts		4 dépôts				6 dépôts					
	D1	D2	D1	D2	D3	D4	D1	D2	D3	D4	D5	D6
<i>LWFR pas long</i> Taux	0,5	0,6	0,8	0,5	0,4	0,6	0,6	0,5	0,6	0,8	0,7	0,8
<i>LWFR pas long</i> <i>S globale</i>	0,55		0,58				0,66					
<i>LWFR pas court</i> Taux	0,2	0,3	0,2	0,1	0,2	0,2	0,2	0,1	0,1	0,2	0,3	0,2
<i>LWFR pas court</i> <i>S globale</i>	0,25		0,15				0,18					

*Tableau 3.2 : résultats du scénario 1.*



**Figure 3.4 :** Influence de nombre de dépôts sur la survie globale

**Discussion et comparaison**

Dans le scénario 1, nous avons étudié l'influence de la variation du nombre de dépôts sur la survie du système pour les deux versions de l'algorithme LWFR (à pas court, à pas long). Le résultat est montré dans le *Tableau 3.2* et la *Figure 3.4*. On remarque dans le cas de LWFR à pas long que la survie de système adopte une augmentation avec l'ajout des nouveaux dépôts dans l'environnement, alors que dans la version LWFR à pas court que la progression de la courbe n'est pas stable est toujours inférieur à celle de LWFR à pas long.

**5.3.2. Scénario 2**

Ce scénario traite l'influence de la taille de l'environnement sur la performance des deux algorithmes LWFR a pas court et LWFR a pas long. Nous avons utilisé les tailles :  $30p \times 30p$ ,  $50p \times 50p$ ,  $80p \times 80p$  et  $100p \times 100p$ .

Taille environnement	$30p \times 30p$				$50p \times 50p$				$80p \times 80p$				$100p \times 100p$			
Dépôts	D1	D2	D3	D4	D1	D2	D3	D4	D1	D2	D3	D4	D1	D2	D3	D4
Algorithme																
LWFR pas long Taux	0,9	0,8	0,9	0,9	0,8	0,5	0,4	0,6	0,8	0,4	0,5	0,7	0,4	0,2	0,5	0,3
LWFR pas long <i>S global</i>	0,88				0,57				0,43				0,35			
LWFR pas court Taux	0,9	0,8	0,8	0,8	0,2	0,2	0,1	0,2	0,2	0,1	0,1	0,1	0,1	0	0,1	0
LWFR pas court <i>S global</i>	0,83				0,18				0,13				0,01			

Tableau 3.3 : Influence de la taille de l'environnement sur les performances (scénario 2)

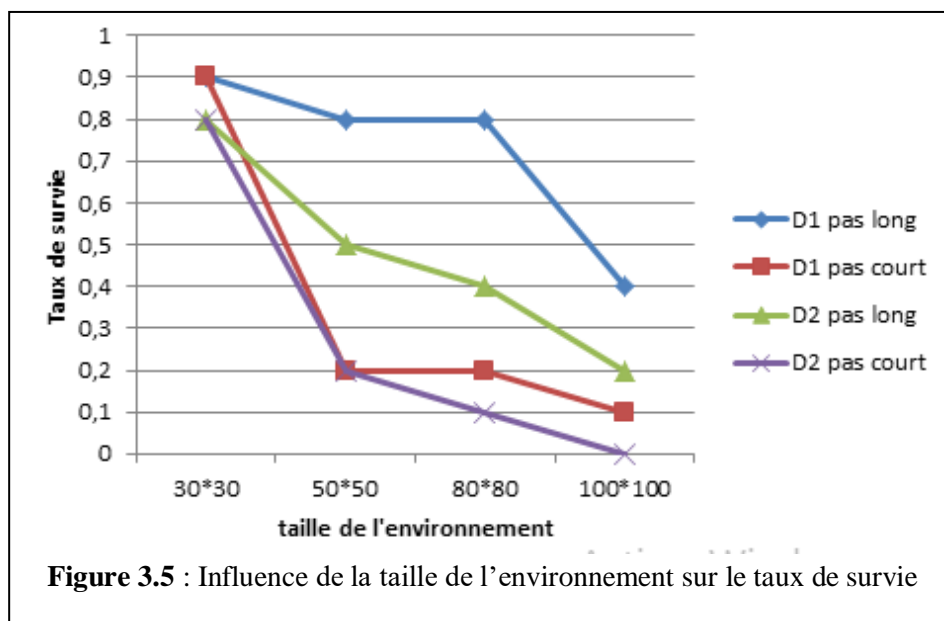


Figure 3.5 : Influence de la taille de l'environnement sur le taux de survie

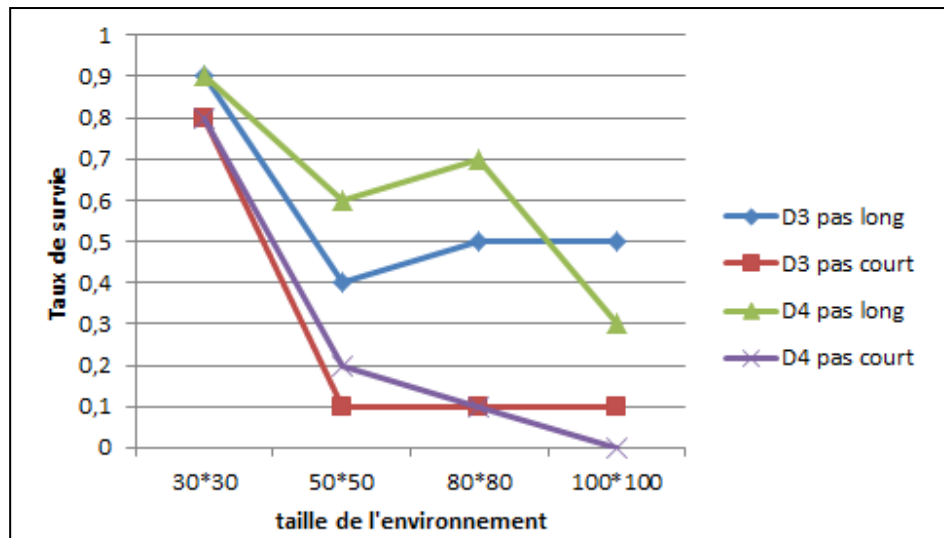


Figure 3.6 : Influence de la taille de l'environnement sur le taux de survie

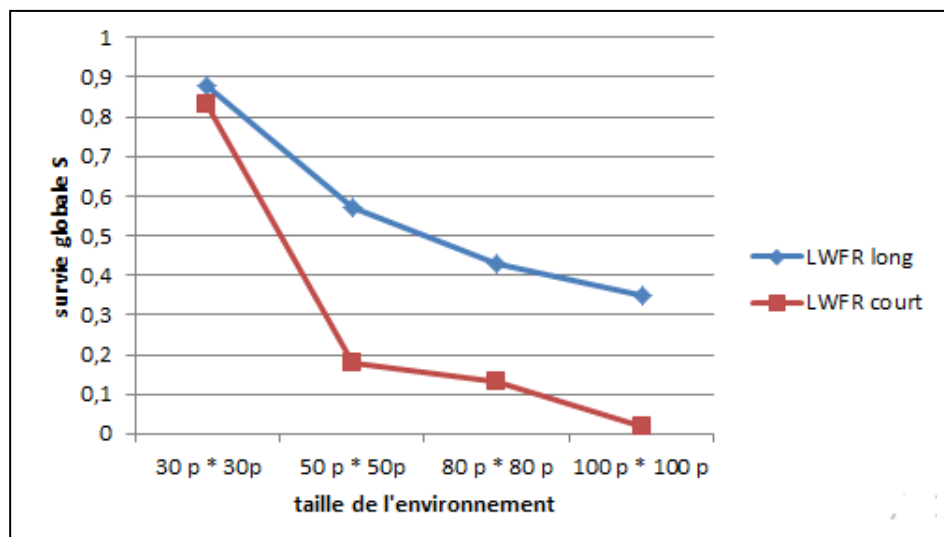


Figure 3.7 : Influence de la taille de l'environnement sur la survie global

### Discussion et comparaison

Dans le scénario 2, nous avons étudié l'influence de la variation de la taille de l'environnement sur la survie du système pour les deux versions de l'algorithme LWFR (à pas court, à pas long). Le résultat est montré dans le *Tableau 3.3* et les *Figures 3.5, 3.6* et *3.7*. On remarque que le pourcentage de survie de système diminue lorsqu'en augmente la taille de l'environnement dans les deux versions de l'algorithme (long, court), cette augmentation de la taille implique que la zone de l'exploitation augmente, donc plus de temps sera consacré

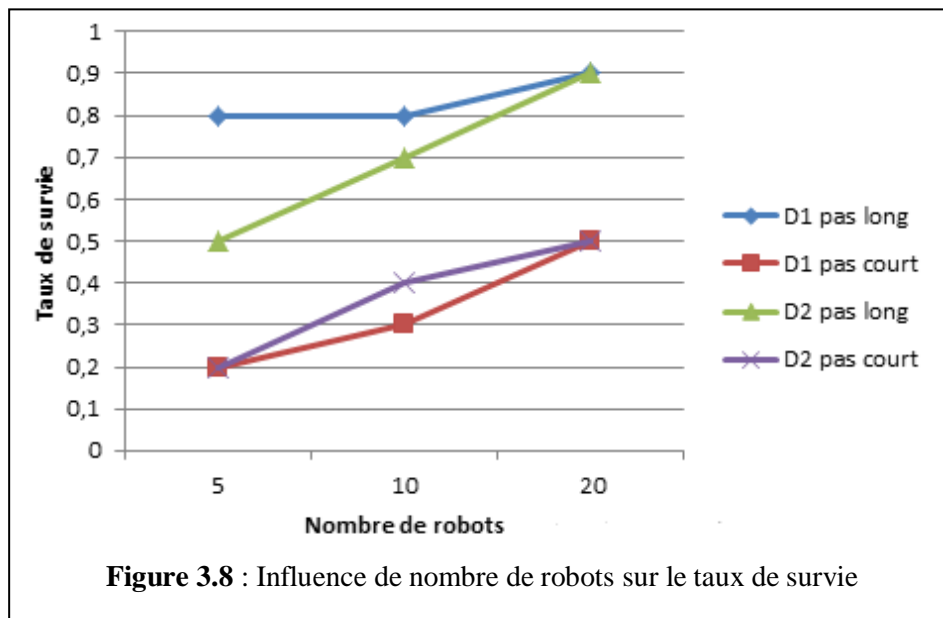
seulement à l'exploration, ce qui diminue le taux de survie du système. Finalement la version LWFR à pas long donne des bons résultats par rapports à la version LWFR à pas court.

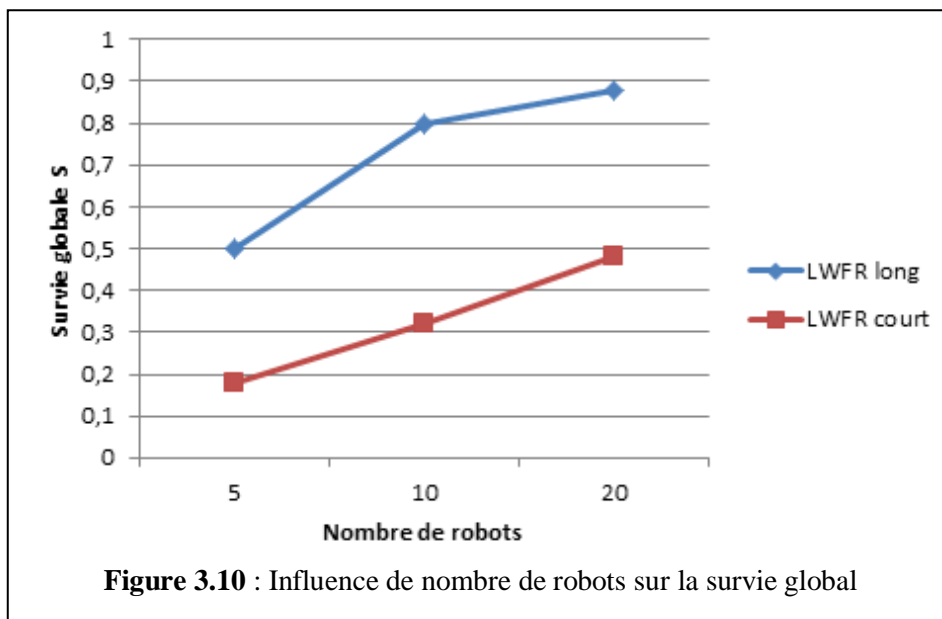
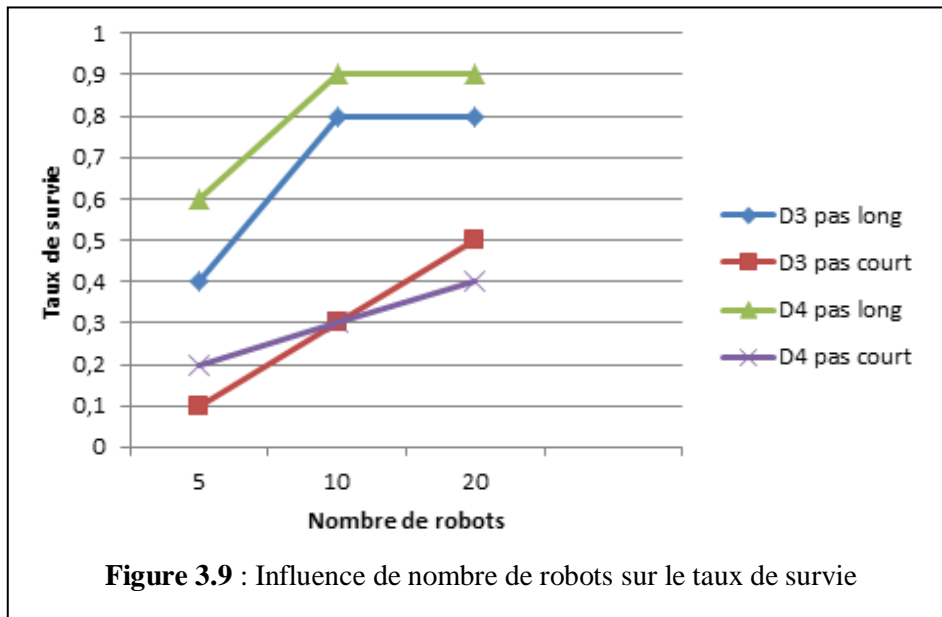
### 5.3.3. Scénario 3

Nous avons étudié dans ce scénario l'influence du nombre de robots sur les performances des deux versions de LWFR, nous avons ainsi augmenté le nombre de robots de 5 jusqu'à 20 robots.

Nombre robots	5 robots				10 robots				20 robots			
Dépôts	D1	D2	D3	D4	D1	D2	D3	D4	D1	D2	D3	D4
Algorithmes												
LWFR pas long Taux	0,8	0,5	0,4	0,6	0,8	0,7	0,8	0,9	0,9	0,9	0,8	0,9
LWFR pas long S global	0,58				0,8				0,88			
LWFR pas court Taux	0,2	0,2	0,1	0,2	0,3	0,4	0,3	0,3	0,5	0,5	0,5	0,4
LWFR pas court S global	0,18				0,32				0,48			

Tableau 3.4 : influence de la variation de nombre de robots sur les performances





### Discussion et comparaison

Dans le scénario 3, nous avons étudié l'influence de la variation du nombre de robots sur la survie du système pour les deux versions de l'algorithme LWFR (à pas court, à pas long). Les *Tableau 3.4* et les *Figure 3.8, 3.9* et *3.10* montrent les résultats obtenus avec l'algorithme LWFR dans les deux cas (pas long, pas court), lorsque le nombre des robots varie de 5-20. Le



pourcentage de survie de système augmente avec toute augmentation du nombre des robots. L'algorithme *LWFR* à pas long est beaucoup mieux que l'algorithme *LWFR* à pas court. *LWFR* à pas long donne toujours les meilleurs résultats.

### 6. Conclusion

Ce chapitre a été consacré à l'implémentation et le test des résultats obtenus par l'algorithme proposé. Nous avons présenté au début, la plateforme Netlogo. Par la suite, nous avons expliqué les composants de notre système ainsi que leur caractéristique. Nous avons ensuite, présenter les paramètres de l'algorithme *LWFR*. Puis, nous avons analysé les performances de *LWFR* dans les deux versions (à pas long et court).

Les résultats obtenus ont été comparés avec la version *LWFR* à pas court. Typiquement, dans toutes les simulations *LWFR* à pas long montre sa supériorité par rapport au *LWFR* à pas court. Ce qui prouve quantitativement que *LWFR* à pas long proposé est efficace.

## Conclusion générale et perspectives

Le problème de Foraging est un problème de référence en robotique mobile. L'étude du foraging est importante pour plusieurs raisons : (1) parce qu'il constitue une métaphore d'une vaste catégorie de problèmes : le nettoyage, la récolte, la recherche et le sauvetage, le déminage ou l'astrobiologie planétaire, (2) c'est une tâche complexe qui inclut l'étude et la coordination de plusieurs autres tâches intégrant l'exploration, la navigation, la manipulation et le transport d'objets, (3) parce que aucun système de foraging n'as été déployé dans le monde réel, tous les travaux sont encore dans les laboratoires de recherche.

Dans le but d'aller à un déploiement réel des systèmes de foraging, de nouveaux travaux de recherche commencent à considérer certaines caractéristiques des applications du monde réel tel que : l'adaptation en ligne, la durabilité du système et l'adaptabilité dans des scénarios critiques. Nous avons encore été orientés vers cette branche de recherche.

Dans ce mémoire, nous avons proposé un algorithme de foraging Multi-dépôts qui permet de maintenir la durabilité de fonctionnement du système. Un système externe extrait de temps en temps les objets déposés dans les différents dépôts et le système de foraging doit répondre aux demandes de ces systèmes, autrement maintenir sa durabilité. Notre proposition fait une hybridation entre les algorithmes bio-inspirés Lévy walk et Firefly. Une hybridation des deux algorithmes produit un algorithme qui combine les avantages des deux afin d'assurer la survie du système.

L'algorithme proposé LWFR a été testé sur plusieurs configurations. Nous avons comparé LWFR à pas long avec la version à pas court. Les résultats de simulations obtenus montrent la supériorité d'algorithme LWFR à pas long par rapport à l'algorithme LWFR à pas court.

Comme perspective à court terme de ce travail, nous allons : améliorer et implémenter l'algorithme LWFR dans une plateforme de simulation robotique (Argos, Gazebo, ...). Alors que, à long terme nous souhaitons appliquer l'algorithme dans le monde réel.

## Bibliographie

**[Baldassarre et al., 2007]** : Baldassarre, G., Trianni, V., Bonani, M., Mondada, F., Dorigo, M., & Nolfi, S. (2007). Self-organized coordinated motion in groups of physically connected robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(1), 224-239.

**[Banerjee and Moses, 2010b]** : Banerjee, S., & Moses, M. (2010). Immune System Inspired Strategies for Distributed Systems. *arXiv preprint*, 301-318.

**[Beni et Wang, 1989]**: Beni, G., & Wang, J. (1989). Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26-30. *Y.: NATO*.

**[Beni, 2005]** : Beni, G. (2005). From swarm intelligence to swarm robotics. vol. 3342 of Lecture Notes in Computer Science, pp. 1–9.

**[Beni, 2004]** : Beni, G. (2004, July). From swarm intelligence to swarm robotics. In International Workshop on Swarm Robotics (pp. 1-9). Springer, Berlin, Heidelberg.

**[Blum et Merkle, 2008]** : Blum, C., & Merkle, D. (2008). Swarm intelligence. Swarm Intelligence in Optimization; Blum, C., Merkle, D., Eds, 43-85.

**[Brambilla et al., 2013]** : Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1), 1-41.

**[Castello et al., 2013]**: Castello, E., Yamamoto, T., Nakamura, Y., & Ishiguro, H. (2013, October). Task allocation for a robotic swarm based on an adaptive response threshold model. In *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*(pp. 259-266). IEEE

**[Castello et al., 2016]**: Castello, E., Yamamoto, T., Dalla Libera, F., Liu, W., Winfield, A. F., Nakamura, Y., & Ishiguro, H. (2016). Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach. *Swarm Intelligence*, 10(1), 1-31.

- [Castello et al., 2014]:** Castello, E., Yamamoto, T., Nakamura, Y., & Ishiguro, H. (2014). Foraging optimization in swarm robotic systems based on an adaptive response threshold model. *Advanced Robotics*, 28(20), 1343-1356.
- [Charrier et al.,2007] :** Charrier, R., Bourjot, C., & Charpillet, F. (2007, November). Un modèle connexionniste pour l'intelligence en essaim: le système multi-agent logistique. In *Colloque de l'Association pour la Recherche Cognitive-ARCo'07: Cognition-Complexité-Collectif*(pp. 19-32).
- [Correll et al., 2008] :** Correll, N., Rutishauser, S., & Martinoli, A. (2008). Experimental Robotics, vol. 39 of Springer Tracts in Advanced Robotics, chap. Comparing Coordination Schemes for Miniature Robotic Swarms : A Case Study in Boundary Coverage of Regular Structures, pp. 471–480.
- [Dorigo et Sahin, 2004] :** Dorigo, M., & Sahin, E. (2004). Guest editorial: Swarm robotics. *Autonomous Robotics*, 17(2-3), 111-113.
- [Garnier, 2008] :** Simon Garnier, « Décisions collectives dans des systèmes d'intelligence en essaim », thèse de doctorat en Ethologie soutenu en 2008, université Toulouse3, 2008.
- [Gasparri et al., 2012] :** Gasparri, A., Oriolo, G., Priolo, A., & Ulivi, G. (2012, October). A swarm aggregation algorithm based on local interaction for multi-robot systems with actuator saturations. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 539-544). IEEE.
- [Gnedenko et al .,1954] :** Gnedenko, B. V., Kolmogorov, A. N., Gnedenko, B. V., & Kolmogorov, A. N. (1954). Limit distributions for sums of independent. *Am. J. Math*, 105.
- [Hayes et al., 2002] :** Hayes, A., Martinoli, A., & Goodman, R. (2002). Distributed odor source localization. *Sensors Journal, IEEE*, 2(3) :260–271.
- [Hecker et Moses,2015]:** Hecker, J. P., & Moses, M. E. (2015). Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms. *Swarm Intelligence*, 9(1), 43-70.
- [Ijspeert et al., 2001] :** Ijspeert, A. J., Martinoli, A., Billard, A., & Gambardella, L. M. (2001). Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Autonomous Robots*, 11(2), 149-171

- [**lu et al., 2016**]: Lu, Q., Hecker, J. P., & Moses, M. E. (2016, October). The MPFA: A multiple-place foraging algorithm for biologically-inspired robot swarms. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3815-3821). IEEE.
- [**lu et al., 2018**] : Lu, Q., Hecker, J. P., & Moses, M. E. (2018). Multiple-place swarm foraging with dynamic depots. *Autonomous Robots*, 42(4), 909-926.
- [**Kashiwagi et al.,2006**] : Kashiwagi, A., Urabe, I., Kaneko, K., & Yomo, T. (2006). Adaptive response of a gene network to environmental changes by fitness-induced attractor selection. *PloS one*, 1(1), e49.
- [**Krieger & Billeter, 2000**] : Krieger, M. J., & Billeter, J. B. (2000). The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, 30(1-2), 65-84.
- [**Kube & Bonabeau, 2000**] : Kube, C. R., & Bonabeau, E. (2000). Cooperative transport by ants and robots. *Robotics and autonomous systems*, 30(1-2), 85-101.
- [**Lévy, 1937**] : Paul, L. (1937). Théorie de l'addition des variables aléatoires. *Gauthiers-Villars, Paris*.
- [**Li et al.,2016**] : Li, W., Gauci, M., & Groß, R. (2016). Turing learning: a metric-free approach to inferring behavior and its application to swarms. *Swarm Intelligence*, 10(3), 211-243.
- [**Martinoli et al., 1999**] : Martinoli, A., Ijspeert, A. J., & Mondada, F. (1999). Understanding collective aggregation mechanisms: From probabilistic modelling to experiments with real robots. *Robotics and Autonomous Systems*, 29(1), 51-63.
- [**McLurkin & Smith, 2007**] : McLurkin, J., & Smith, J. (2007). Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. In *Distributed autonomous robotic systems 6* (pp. 399-408). Springer, Tokyo.
- [**Reynolds, 1987**] : Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model (Vol. 21, No. 4, pp. 25-34). ACM.
- [**Schwager et al., 2006**] : Schwager, M., McLurkin, J., & Rus, D. (2006, August). Distributed Coverage Control with Sensory Feedback for Networked Robots. In *robotics: science and systems* (pp. 49-56).

[**Yang XS, 2009**] : Yang, X. S. (2009, October). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms* (pp. 169-178). Springer, Berlin, Heidelberg.

[**Yang et al., 2009**] : Yang, Y., Zhou, C., & Tian, Y. (2009, February). Swarm robots task allocation based on response threshold model. In *2009 4th International Conference on Autonomous Robots and Agents*(pp. 171-176). IEEE.

[**Winfield,2009**] : Winfield, A. F. (2009). Foraging robots. *Encyclopedia of complexity and systems science*, 3682-3700.

[**Winfield et al., 2010**] : Winfield, A. F., Kernbach, S., & Schmickl, T. (2010). Collective foraging: cleaning, energy harvesting and trophallaxis.351-409.

[**Wilensky, 1999**] : Wilensky, U. (1999). NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

### Webographie

[1] : Banc de poisson, URL : <https://bionique.artbite.fr/S-organiser-en-bancs-compacts.html>, dernière consultation : 08/07/2019.

[2] : Essaim d'oiseaux, URL : <https://blogs.futura-sciences.com/lehning/2018/09/24/les-vols-detourneaux>, dernière consultation : 08/07/2019.

[3] : Essaim d'abeille, URL : <http://www.trigobert.net/essaimage-dabeilles-focus-sur-la-duplication-de-la-colonie.html>, dernière consultation : 08/07/2019.

[4] : Essaim de fourmis, URL : <https://fr.depositphotos.com/130175886/stock-photo-swarm-of-ants-eating-dicarded.html>, dernière consultation : 08/07/2019.

[5] : Fireflies lightning, URL : <https://www.sciencenews.org/article/fireflies-lightning-bugs-flashes-predators>, dernière consultation : 08/07/2019.