

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de L'Enseignement Supérieur et de la Recherche Scientifique



UNIVERSITÉ FERHAT ABBAS - SETIF1

FACULTÉ DES SCIENCES

THÈSE

Présentée au Département d'informatique

Pour l'obtention du diplôme de

DOCTORAT

Domaine : Sciences et Technologie

Filière: informatique

Option: informatique

Par

RABHI Seddik

THÈME

**Optimisation des algorithmes de localisation dans
les réseaux de capteurs sans fil**

Soutenue le/...../..... devant le Jury:

LAKHFIF Abdelaziz	MCA	Univ. Ferhat Abbas Sétif 1	Président
SEMCHEDINE Fouzi	Professeur	Univ. Ferhat Abbas Sétif 1	Directeur de thèse
AMAD Mourad	MCA	Univ. de Bouira	Examineur
BADACHE Abderrahmane	MCA	Univ. d'Alger	Examineur

Remerciements

Avant tout, je remercie en premier lieu Dieu, le Tout Puissant, de m'avoir donné autant de courage, de volonté et de patience pour pouvoir achever ce travail, qu'il soit béni et glorifié.

*J'exprime ma profonde gratitude à Mr. **Fouzi SEMCHEDINE**, Professeur à l'université Ferhat Abbas Sétif 1, pour m'avoir accordé toute sa confiance en acceptant de m'encadrer. Pour ses conseils et son aide dans la réalisation de notre travail. Je tiens également à rendre hommage à sa rigueur et à sa disponibilité. Et pour finir, je ne cesserai de la remercier pour ses précieux conseils, qui ont été d'une grande utilité pour l'achèvement en bon terme de ce travail.*

*Je tiens tout d'abord à remercier Mr. **Abdelaziz LAKHIFIF**, Maître de conférences à l'université Ferhat Abbas Sétif 1, pour l'honneur qu'il me fait l'honneur d'accepter d'évaluer de juger ce travail et d'en être président de jury de cette thèse.*

*Je remercie également Mr. **Mourad AMAD** Maître de conférences à l'université de bouira et Mr **Abderrahmane BADACHE** Maître de conférences à l'université d'alger. D'avoir accepté de lire et d'évaluer ce travail ceci malgré un emploi du temps certainement très chargé.*

*D'autre part, Je voudrais remercier toutes les personnes avec qui j'ai eu la chance d'avoir d'enrichissants échanges scientifiques. En particulier, Mr. **Nader MBAREK**, Maître de conférences, Habilité à Diriger des Recherches à l'Université de Bourgogne Franche-Comté Dijon-France. Pour m'avoir accueillie dans son laboratoire pour un stage long duré dans le cadre de programme « PNE » de notre département ministérielle. Ses qualités humaines et scientifiques sont pour moi une source constante d'inspiration.*

L'amour et la patience dont a fait preuve Je remercie en notamment ma famille qui a toujours été là lorsque j'en avais besoin et je lui suis grandement reconnaissant. Que tous ceux qui ont, de près ou de loin, apporté une aide à ce travail, trouvent ici l'expression de mes remerciements les plus sincères

Dédicaces

Je dédie ce modeste travail

À mon père et à ma mère qui m'ont soutenu et encouragé dans les moments les plus difficiles...

À mes frères et à mes sœurs ...

À ma femme et à mes enfants « Abdelhafidh » et « Ahmed Yassine » leur patience et leur affection...

À toute ma famille et à tous mes amis...

Résumé

Les travaux présentés dans cette thèse s'inscrivent dans le cadre général des réseaux de capteurs sans fil. Elles portent sur l'étude et la mise en œuvre de nouveaux algorithmes distribués qui utilisent les méta-heuristiques bio-inspirées pour résoudre le problème de localisation. Dans cette thèse, nous adaptons la méta-heuristique de l'algorithme d'optimisation par Fruit Fly (FOA) pour optimiser la carte des positions estimées par deux célèbres techniques de localisation existantes (DV-HOP et DV-DISTANCE) afin de fournir deux nouvelles approches hybrides de localisation. Dans un second lieu, un nouvel algorithme de localisation basé seulement sur la méta-heuristique FOA est proposé. Pour valider les performances de nos approches proposées, nous avons effectué des expérimentations et nous les avons comparés avec les méthodes traditionnelles de DV-hop et DV-distance pour évaluer le taux d'amélioration. Par ailleurs, la précision et le temps d'exécution de la nouvelle méthode de localisation ont été évalués et comparés avec ceux d'autres méta-heuristiques de localisation performantes.

Mots clés : réseaux de capteurs sans fil, localisation, méta-heuristiques, optimisation, Fruit Fly.

Abstract

The work presented in this thesis falls within the general framework of sensor networks. They relate to the study and implementation of new distributed algorithms using bio-inspired meta-heuristics to resolve the problem of localization. In this thesis, we adapt the meta-heuristic of the Fruit Fly optimization algorithm (FOA) to optimize the position map estimated by two famous existing localization techniques (DV-HOP and DV-DISTANCE) in order to provide two new localization approaches. In the second place, a new localization algorithm based only on the FOA meta-heuristic is proposed. To validate the performances of our proposed approaches, we carried out experiments which we compare with the traditional methods of DV-hop and DV-distance to have the rate of improvements in the two approaches. Further, the precision and the execution time of the new bio-inspired method have been evaluated and compared with those of efficient localization meta-heuristics.

Keywords: wireless sensor networks, localization, meta-heuristics, optimization, Fruit Fly.

Sommaire

Remerciements	i
Dédicaces	ii
Résumé	iii
Sommaire	iv
Liste des figures	v
Liste des Tableaux	vi
Introduction générale.....	01

Chapitre I : Introduction aux réseaux de capteurs sans fil

I.1 Introduction	05
I.2 Le capteur	05
I.2.1 Définition d'un capteur	05
I.2.2 Composition d'un capteur	06
I.2.3 Types de captures	07
I.3 Les réseaux de capteurs sans fils.....	08
I.3.1 Définition d'un RCSF.....	08
I.3.2 Le déploiement des nœuds du RCSF	09
I.3.3 Caractérisation des RCSFs.....	09
I.4 Domaines d'application des RCSFs.....	11
I.5 Défis des réseaux de capteurs sans fil	13
I.6 Conclusion.....	17

Chapitre II : Problème de localisation dans les RCSFs

II.1 Introduction	19
II.2 Définition de la localisation	19
II.3 Système de localisation	20
II.3.1 Critères d'un système de localisation	20
II.3.2 Caractérisations d'un système de localisation.....	21
II.3.2.1 Système des coordonnées	21
II.3.2.2 Forme d'implémentation	21
II.3.2.3 Utilisation d'estimations de distances	22
II.3.2.4 Nécessité des ancres	22
II.3.3 Phases principale d'un système de localisation	22
II.4 Modèle d'estimation des distances	24
II.4.1 Le RSSI (Received Signal Strength Indicator).....	26
II.4.2 Le LQI (Link Quality Indicator).....	26
II.4.3 ToA/ToF (Time of Arrival, Time of Flight).....	27
II.4.4 TDoA (Time Difference of Arrival).....	27
II.4.5 AoA/DoA (Time Difference of Arrival).....	28
II.4.6 Bilan des techniques de mesures.....	28
II.5 Mécanismes de dérivation des positions.....	29

II.5.1 Multilatération.....	29
II.5.2 Trilatération.....	31
II.5.3 Triangulation	31
II.5.4 Bilan sur les méthodes de dérivation de positions	32
II.6 Un aperçu des algorithmes de localisations existantes.....	33
II.6.1 Bounding Box (Cadre englobant)	33
II.6.2 GPS-Less (centroïde).....	34
II.6.3 APIT	35
II.6.4 MDS-MAP	36
II.6.5 SumDistMinMax.....	37
II.6.6 DV-Euclidien.....	38
II.6.7 DV- Hop.....	39
II.6.8 DV-Distance.....	42
II.6.9 Bilan sur les caractéristiques des algorithmes de localisation	43
II.7 Conclusion.....	44

Chapitre III: Des méta-heuristiques d'optimisation

III.1 Introduction.....	47
III.2 Quelques notions générales sur l'optimisation	48
III.3 Littérature des méta-heuristiques d'optimisation existantes.....	50
III.3.1 Le recuit simulé.....	52
III.3.2 Les algorithmes génétiques	53
III.3.3 Optimisation par des colonies de fourmis	54
III.3.4 L'optimisation par des essaims de particules	55
III.3.5 Algorithme de l'Evolution Différentielle	56
III.3.6 Algorithme de la recherche coucou.....	58
III.3.7 Algorithme des colonies d'abeilles	60
III.3.8 Algorithme de Firefly.....	62
III.3.9 Optimisation par essaim de poulets	63
III.3.10 Algorithme d'optimisation par Fruit Fly	66
III.3.10.1 Aspect biologique	66
III.3.10.2 Modèle artificielle	67
III.4 Conclusion.....	70

Chapitre IV : APPROCHES PROPOSÉES

IV.1 Introduction	72
IV.2 Optimisation des techniques de localisations existantes par FOA	72
IV.2.1 Travaux similaire	72
IV.2.2 Formulation de Problème	75
IV.2.3 DV-HOP(FOA) : Optimisation de DV-HOP par FOA	76
IV.2.3.1 Présentation	76
IV.2.3.2 Résultats de simulation de DV-Hop (FOA).....	78
IV.2.3.3 Récapitulatif.....	82
IV.2.4 DV-Distance(FOA) : Optimisation de DV- Distance par FOA	83
IV.2.4.1Présentation	83
IV.2.4.2Résultats de simulation de DV- Distance (FOA).....	85

Sommaire

IV.2.4.3Récapitulatif.....	88
IV.3 FOA-L : Une nouvelle méta-heuristique de localisation	88
IV.3.1Travaux similaire	88
IV.3.2 Présentation de FOA-L	90
IV.3.3 Résultats de simulation de FOA-L	95
IV.3.3.1 Evaluation de la précision de localisation dans FOA-L....	96
IV.3.3.2 Comparaison de FOA-L avec des travaux similaires.....	98
IV.3.4 Récapitulatif.....	99
IV.4 Conclusion.....	100
Conclusion générale.....	102
Contribution.....	104
Référence bibliographie.....	105

Liste des figures

Chapitre I

Figure I.1 : Fonctionnement d'un capteur.	06
Figure I.2 : Les composants d'un capteur.	06
Figure I.3 : Quelque capteur existant	08
Figure I.4 : Architecture générale d'un RCSF	08
Figure I.5 : Quelques applications des RCSFs.....	13

Chapitre II

Figure II.1 : Phases d'un système de localisation.....	24
Figure II.2 : Exemple de collecte d'informations de voisinage.....	25
Figure II.3 : Principe du TDoA.....	27
Figure II.4 : Principe de la méthode AOA	28
Figure II.5 : Principe de la trilatération.....	31
Figure II.6 : Principe de la triangulation	32
Figure II.7 : Principe du cadre englobant.....	34
Figure II.8 : Exemple de localisation par centroïde	35
Figure II.9 : Principe de la méthode APIT.....	35
Figure II.10 : Principe de SumDist	37
Figure II.11 : Principe de MinMax	38
Figure II.12 : Principe de l'algorithme DV-Euclidien.....	39
Figure II.13 : Exemple de l'algorithme DV-Hop	41
Figure II.14 : Principe de la méthode DV-Distance	43

Chapitre III

Figure III.1 : Point singuliers d'une fonction objective	49
Figure III.2 : Classification des méthodes d'optimisation	49
Figure III.3 : Algorithme de recuit simulé	52
Figure III.4 : Principe de l'Algorithme génétique	53
Figure III.5 : Algorithme de colonies de fourmis	54
Figure III.6 : Principe des algorithmes évolutionnaires	56
Figure III.7 : Théorie de Darwin	56
Figure III.8 : Algorithme de l'évolution différentielle	57
Figure III.9 : Pseudo code de l'Algorithme de recherche coucou	59
Figure III.10: Pseudo code de l'Algorithme ABC	61
Figure III.11: pseudo code de l'Algorithme de Firefly.....	62
Figure III.12: pseudo code de l'Algorithme de CSO	66
Figure III.13: Structure corporelle de la drosophile	67
Figure III.14: Schéma de recherche de la drosophile	68
Figure III.15: Organigramme de l'algorithme	69

Chapitre IV

Figure IV.1 Organigramme de l'algorithme DV-Hop(FOA).....	78
Figure IV.2 Exemple de simulation d'un RCSF avec 50 nœuds placés aléatoirement dans un champ 2D.....	79
Figure IV.3 Erreur de localisation pour chaque nœud inconnu.....	79
Figure IV.4 Moyenne d'erreur par rapport à la variation du nombre de nœuds.....	80
Figure IV.5 Moyenne d'erreur de localisation par rapport à la variation de nombre d'ancres.....	81
Figure IV.6 Moyenne d'erreur de localisation par rapport à la variation de la taille de la population.....	82
Figure. IV.7 Taux d'erreur de localisation par rapport à la superficie de la zone de RCSF.....	85
Figure IV.8 Taux d'erreur de localisation par rapport au rayon de communication.....	86
Figure IV.9 Taux d'erreur de localisation par rapport au Taux des ancres.....	86
Figure IV.10 Recherche d'un nœud cible par FOA-L.....	90
Figure IV.11 Machine à états finis de capteurs inconnus pour l'algorithme FOA-L	93
Figure IV.12 Diagramme de séquence des messages échangés entre les capteurs pour FOA-L.....	94
Figure IV.13 Erreur de localisation pour chaque nœud inconnu.....	96
Figure IV.14 Moyenne d'erreur de localisation par rapport au nombre de capteurs.....	96
Figure IV.15 Moyenne d'erreur de localisation par rapport au nombre d'ancres....	97
Figure IV.16 Moyenne d'erreur de localisation par rapport à la taille de la population.....	98

Liste des Tableaux

Tableau II.1	Comparaison des techniques d'estimation des distances	29
Tableau II.2	Comparaison des méthodes de dérivation de positions	33
Tableau II.3	Valeurs illustratives pour l'exemple de DV-HOP	41
Tableau II.4	Caractéristiques des algorithmes de localisation dans les RCSFs	44
Tableau IV.1	Moyenne d'erreur de DV-Hop (FOA) et DV-Hop.....	80
Tableau IV.2	Paramètres de configuration du scénario de simulation.....	99
Tableau IV.3	Résultats de la comparaison.....	99

Introduction générale

Les capteurs sont utilisés pour collecter des données environnementales telles que la température, la pression, l'humidité, la radiosité, etc. La prévision précoce de tels événements aide à planifier un système de réponse adéquat qui peut soit prévenir ces événements, soit atténuer les dommages consécutifs. Le système de réponse devrait avoir la capacité d'extraire le contexte des données sensorielles recueillies s'il devait prévoir correctement les événements et fonctionner au niveau d'efficacité attendu. Un contexte est défini par des ambiances parmi lesquelles la localisation (à la fois spatiale et temporelle) est l'ingrédient le plus critique.

Les réseaux de capteurs sans fil (RCSFs) dotés de capacités de détection peuvent rassembler des paramètres vitaux liés à la sécurité tels que les communications radio, les signes d'activités accélérées ou les mouvements vigoureux dans une zone pour aider à développer une réponse de sécurité et un système d'alerte avancé. Mais ces paramètres détectés sont inutiles à moins qu'ils ne soient accompagnés d'informations de localisation correspondantes. Bien que les informations de localisation puissent être alimentées manuellement, elles peuvent ne pas être réalisables dans une grande zone de déploiement. Il est donc nécessaire de développer un système capable de détecter automatiquement les informations de localisation de tous les nœuds de la zone de déploiement.

Une solution de localisation consiste à équiper chaque capteur par récepteur GPS. Les capteurs peuvent se localiser individuellement à l'aide des signaux GPS. Cependant, l'installation d'un récepteur GPS pour chaque nœud de capteur augmente considérablement le coût total du réseau de capteurs. De plus, l'introduction du récepteur GPS augmente la consommation d'énergie d'un nœud et raccourcit ainsi sa durée de vie. Ces raisons et diverses autres contraintes, n'ont pas permis l'utilisation des systèmes de localisation existants, tels que le GPS, pour la localisation des nœuds dans les réseaux de capteurs sans fil. Pour permettre aux nœuds d'estimer leur position, de nombreux algorithmes et méthodes ont été engagés.

Les techniques de localisation automatiques existantes utilisent plusieurs voies technologiques. Elles sont très différentes les unes des autres et répondent en général à des besoins différents. Cependant, l'erreur entre la position réelle et la position estimée obtenue par ces méthodes ne pouvait pas être négligé. Par conséquent, il est donc nécessaire de concevoir et de développer de nouvelles stratégies et d'algorithmes pour localiser avec la meilleure précision possible, tous les nœuds du réseau.

Dans ces dernières années, une évolution croissante du nombre d'études menées sur les animaux vivants en groupe ou en société et plus particulièrement les insectes sociaux. Ces études dans la théorie de l'auto-organisation ont inspiré un grand nombre des chercheurs pour développer une nouvelle approche appelée : la méta-heuristique. Les méta-heuristiques sont des algorithmes généraux d'optimisation applicables à une grande variété de problèmes. Elles sont apparues dans le but de résoudre au mieux des problèmes d'optimisation. L'application des méta-heuristiques à la localisation dans les RCSF a connu une évolution particulière ces dernières années grâce aux capacités de trouver des solutions prometteuses et optimales avec le moindre coût.

Dans le cadre des travaux de cette thèse, nous avons donc cherché à mettre en œuvre de nouveaux algorithmes distribués en utilisant les méta-heuristiques bio-inspirées permettant d'apporter des réponses aux nombreuses problématiques posées de la localisation dans les RCSFs. Nous nous sommes intéressés de près à l'optimisation de la localisation. Pour cela, nous allons utiliser une nouvelle méthode Bio-inspirée. Cette approche, appelée Algorithme d'Optimisation par Fruit Fly (FOA), est une méta-heuristique d'optimisation qui imite les capacités de la drosophile à capturer de manière sensible les odeurs par son organe afin de rechercher leur nourriture (fruits).

En effet, l'algorithme d'optimisation FOA est adapté pour optimiser deux techniques de localisation existantes (DV-HOP et DV-DISTANCE) d'un côté, et de l'autre côté un nouvel algorithme de localisation basée seulement sur la méta-heuristique FOA est proposé. Pour valider les performances de nos approches proposées, nous avons effectué des expérimentations que nous comparons avec les méthodes traditionnelles de DV-hop et DV-distance pour avoir le taux d'améliorations dans les deux approches. Par ailleurs, la précision et le temps d'exécution de la nouvelle méthode bio-inspirée ont été comparés avec d'autres méta-heuristiques de localisation performantes.

Pour mener à bien notre contribution, la thèse commence par la présente introduction et s'organise en quatre chapitres:

- Le premier chapitre s'intitule "Introduction aux réseaux de capteurs sans fil". Dans ce chapitre, nous décrivons la définition, les caractéristiques et l'architecture générale d'un réseau de capteurs sans fil. Nous avons également présenté dans ce chapitre : les contraintes de conception d'un réseau de capteurs, les différents domaines d'applications ainsi que les problématiques liées à ce type de réseaux.
- Dans le second chapitre " Problème de localisation dans les RCSFs", nous nous concentrons au vif de notre sujet qui est la localisation dans les réseaux de capteurs sans fil, et nous détaillons les notions d'un système de localisation, leurs propriétés ainsi que les différentes parties qui le composent de tel qu'un système critique pour le développement des RCSFs. De plus, une synthèse exhaustive, critique et comparative des algorithmes de localisation est présentée.
- Le troisième chapitre s'intitule "Des méta-heuristiques d'optimisation". En premier lieu, il présente la notion d'optimisation et ses différentes catégories et se focalise en dernier lieu sur le domaine Bio-inspiré et passe en revue les différents algorithmes inhérents à ce domaine pour la résolution des problèmes d'optimisation.
- Le dernier chapitre présente en détails la conception et la mise en œuvre de nouvelles approches proposées dans le cadre des travaux de cette thèse. Les résultats des simulations obtenus prouvent des améliorations remarquables à la précision des algorithmes existants de localisation dans les RCSFs.

Enfin, nous terminons par une conclusion générale résumant les points essentiels qui ont été abordés ainsi que des perspectives que nous souhaitons accomplir prochainement.

Chapitre I

Introduction aux réseaux de capteurs sans fil

Chapitre I

Introduction aux réseaux de capteurs sans fil

I.1 Introduction

Les réseaux de capteurs sans fil (RCSFs) sont une nouvelle classe des réseaux qui deviennent très populaires avec un grand nombre d'applications civiles et militaires. Le RCSF est un réseau sans fil qui contient des dispositifs (capteurs) indépendants et distribués destinés à surveiller les conditions physiques ou environnementales. Un réseau de capteurs sans fil se compose d'un ensemble de nœuds capteurs minuscules, qui communiquent entre eux et échangent des informations et des données. Ces nœuds obtiennent des informations sur l'environnement telles que la température, la pression, l'humidité ou les polluants et envoient ces informations à une station de base. Cette dernière envoie les informations à un réseau filaire ou active une alarme ou une action, selon le type et l'ampleur des données surveillées [Deng et al, 2008] [Sohraby et al, 2007].

Les applications typiques incluent la surveillance du temps et des forêts, la surveillance du champ de bataille, la surveillance physique des conditions environnementales, telles que la pression, la température, les vibrations, les polluants ou le traçage des mouvements humains et animaux dans les forêts et les frontières [Deng et al, 2008] [Dargie et Poellabauer, 2010].

L'objectif de ce chapitre est de fournir une présentation générale des réseaux de capteurs sans fil, leurs applications et caractéristiques, de la taxonomie et des défis de ce type de réseau.

I.2 Le capteur

I.2.1 Définition d'un capteur

Un capteur est un dispositif qui transforme l'état d'une grandeur physique (telles que la température, les vibrations et la pression) observée en une grandeur utilisable comme le

montre la Figure I.1. Cet appareil est augmenté de capacités de calcul et de communication avec d'autres nœuds afin de faire circuler les informations détectées à travers un réseau sans fil [Deng et al, 2008].

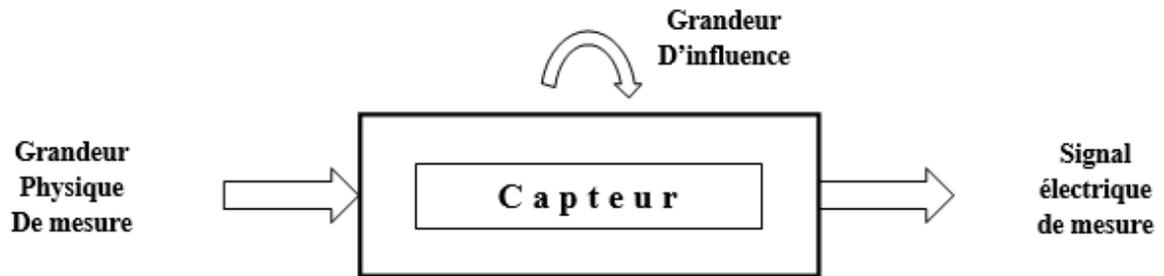


Figure I.1 : Fonctionnement d'un capteur

I.2.2 Composition d'un capteur

Un nœud capteur est un assemblage de quatre composants principaux représentés sur la Figure I.2 : unité de capture, une unité de traitement, une unité de communication et une unité d'énergie ainsi que trois autres composants additionnels sont envisageables selon le domaine d'application tels qu'un système de localisation, un mobilisateur chargé de déplacer le micro-capteur en cas de nécessité et un générateur d'énergie (cellule solaire) [Su et al, 2001].

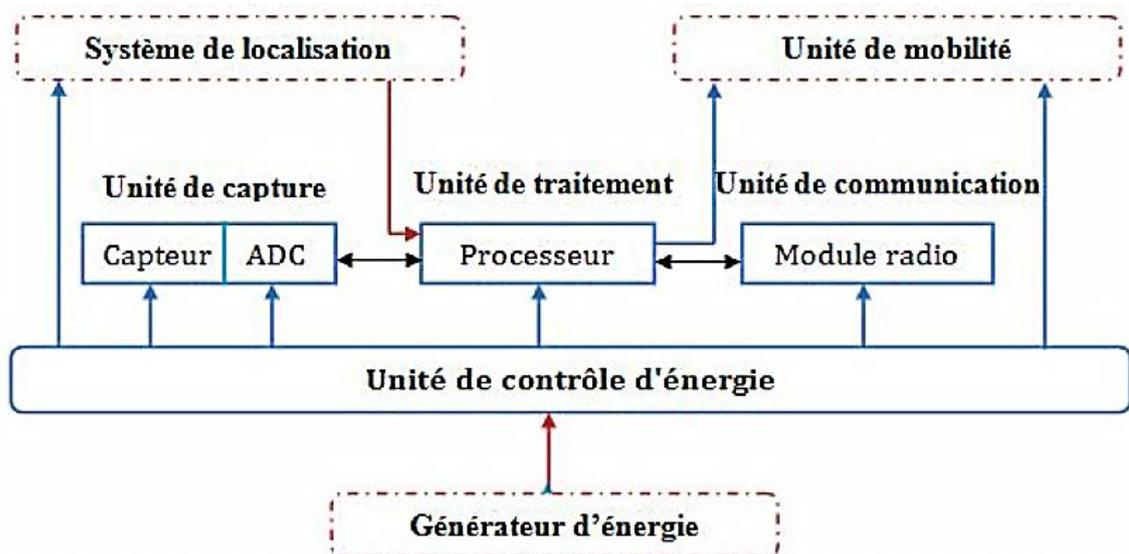


Figure I.2 : Les composants d'un capteur.

On peut voir sur la Figure I.2 les différents composants qui constituent un nœud capteur. Pour être plus précis chaque partie des composants possède son propre rôle :

- **Unité de capture** : Elle permet de capter le phénomène observé et le convertir depuis un signal analogique en un numérique. Il sera, ensuite, fourni à l'unité de calcul.
- **Unité de traitement** : C'est l'unité principale du capteur, elle est chargée d'exécuter les protocoles de communication qui permettent aux nœuds capteurs de collaborer avec les autres nœuds du réseau afin d'accomplir la requête en question.
- **Unité de communication** : Elle a pour fonction l'exécution de toutes les émissions et les réceptions des données.
- **Unité de contrôle d'énergie** : Elle effectue des opérations de contrôle de l'énergie restante et de mesure de la durée de vie du capteur.
- **Système de localisation** : Il fournit des informations sur la position du nœud capteur.
- **Unité de mobilité** : Selon le type du réseau, il est appelé si le capteur doit être déplacé afin d'accomplir sa tâche.
- **Générateur d'énergie** : Il récupère l'énergie de l'environnement extérieur tel que l'énergie solaire puisque le capteur se laisse pendant des mois et même durant des années sans l'intervention humaine. Celle-ci n'est pas un grand problème pour les réseaux sans fil traditionnel, car on peut toujours recharger les batteries des dispositifs sans fil comme les téléphones portables ou les ordinateurs portables. Mais, dans un RCSF, il est difficile (parfois impossible dans certaines applications) de changer la batterie [Lin et al, 2005].

I.2.3 Types de captures

Tout dépend de l'application à laquelle ils sont destinés, les capteurs sont disponibles en différents modèles. Il existe plusieurs fabricants de capteurs parmi lesquels nous trouvons : Imote2, Art of Technology et Crossbow [Raghunathan et al, 2002].

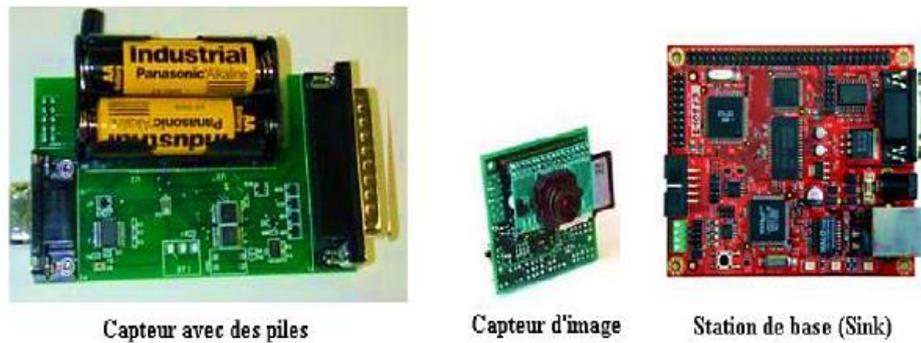


Figure I.3 : Quelques capteurs existant

I.3 Les réseaux de capteurs sans fil

I.3.1 Définition d'un RCSF

Un réseau de capteurs peut être vu comme un réseau de microsystèmes autonomes disséminés dans une zone géographique appelée zone de captage ou zone d'intérêt et communiquent entre eux via une liaison sans fil. Les éléments de ce réseau consistent en un nombre très important de capteurs capables de récolter et de transmettre des données environnementales au nœud passerelle (puits ou en anglais "sink") d'une manière autonome, et mettant en œuvre un routage jusqu'au nœud considéré comme un point de collecte [Dargie et Poellabauer, 2010].

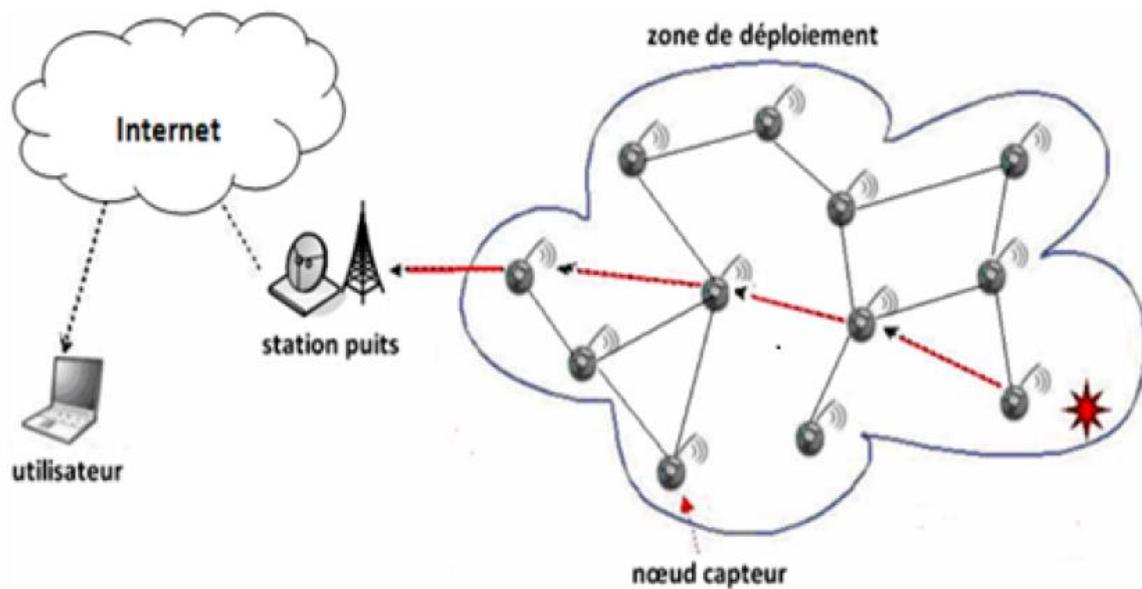


Figure I.4 : Architecture générale d'un RCSF

I.3.2 Le déploiement des nœuds du RCSF

Le déploiement constitue la première étape de la mise en place d'un réseau de capteurs sans fil. C'est une étape, tout comme l'analyse des informations collectées par la station de base, qui nécessite une intervention humaine [CHALLAL, 2008]. Ce déploiement peut se faire de deux manières :

- **Déterministe** : c'est-à-dire que la position des capteurs est bien déterminée et donc connue à l'avance.
- **Aléatoire** : dans ce cas, les capteurs s'auto organisent et utilisent des stratégies de *localisation* pour former le réseau.

I.3.3 Caractérisation des RCSFs

Les capteurs en réseau offrent un nouveau moyen de surveillance et d'autres applications tactiques. Bien que les réseaux de capteurs pour différentes applications puissent être très différents, ils partagent des caractéristiques communes. Les capteurs sont principalement des appareils électriques, électroniques ou électromécaniques, même s'il existe d'autres types de capteurs. En général, un capteur est un type de transducteur qui convertit une entrée en une autre forme, généralement électrique. Les capteurs peuvent être directs ou couplés. Un exemple de capteur direct est un thermomètre ou un compteur électrique qui indique directement. Un capteur couplé utilise un convertisseur analogique-numérique (A/D) afin de convertir un signal analogique en un signal numérique. Les capteurs sont souvent utilisés dans des applications telles que la médecine, l'industrie, l'environnement, la robotique et l'armée. Avec les progrès de la technologie des matériaux, de plus en plus des capteurs sont construits avec la technologie Micro-Electro-Mechanic-Systems (MEMS). Un bon capteur/transducteur devrait avoir les principales caractéristiques suivantes [Dhurandher et al, 2008] [Sohraby et al, 2007] :

- (1) Il doit être sensible à la propriété considérée.
- (2) Il doit être insensible à toute autre propriété.
- (3) Il est souhaitable que le signal de sortie du capteur soit exactement proportionnel à la valeur de la caractéristique mesurée.
- (4) Il devrait avoir une durée de vie raisonnable.
- (5) Il ne doit pas consommer beaucoup d'énergie.

Un RCSF est composé de centaines, voire de milliers de nœuds qui utilisent des dispositifs de détection (capture) pour observer différentes conditions et environnements, tels que le mouvement, la pression, la température, le son, les vibrations, la pollution, les niveaux d'oxygène ou de dioxyde de carbone, l'intensité du trafic et modèles, entre autres, sur différents sites. En général, ces appareils sont minuscules et peu coûteux, ils peuvent donc être fabriqués et déployés en grande quantité. Une différence majeure entre les réseaux traditionnels (MANET, Mobile Ad-hoc NETWORKS) et les RCSF est que les RCSF ont souvent des ressources strictement limitées en termes d'alimentation, de mémoire, de puissance de calcul et de bande passante. Le nœud capteur est une unité autonome équipée d'un émetteur-récepteur radio, d'un minuscule microcontrôleur et d'une source d'alimentation qui est généralement une batterie.

Les nœuds auto-organisent dynamiquement leur configuration en fonction des différentes circonstances du réseau. En raison de la durée de vie limitée des batteries, les nœuds sont construits dans un souci d'économie d'énergie et passent généralement de grandes quantités de temps en mode «veille» ou en mode transmission des données. Par conséquent, chaque capteur est équipé d'une capacité de communication sans fil et de capacités de traitement du signal et de mise en réseau.

Les fonctions principales de tout RCSF sont la détection, la communication et l'informatique [Deng et al, 2008] [Nicopolitidis et al, 2003]. Un schéma pour catégoriser les réseaux de capteurs sans fil est basé sur le fait que les nœuds sont adressables séparément, et un autre est basé sur le fait que les données du réseau sont agrégées. Par exemple, le nœud capteur dans un réseau de parking devrait être adressable individuellement, afin que l'on puisse trouver les endroits de tous les espaces libres. Toutefois, si une personne veut connaître la température ou la pression dans une zone d'angle spécifique alors l'adressage peut ne pas être si important.

La capacité du RCSF à combiner les données collectées peut considérablement réduire le nombre de messages à envoyer via le réseau. Dans certaines situations, il est essentiel d'envoyer le signal par le capteur en temps opportun, par exemple lorsqu'il est nécessaire d'envoyer un signal d'alerte de données à la police indiquant qu'un intrus tente d'entrer dans la maison ou le bureau d'une personne.

I.4. Domaines d'application des RCSFs

La recherche dans le domaine des RCSFs subit actuellement une révolution importante et leurs applications prennent de plus en plus d'ampleur dans plusieurs domaines (voir Figure I.5). Nous citons ici quelques domaines très importants :

- **Militaire** : c'est le moteur initial pour le développement des réseaux de capteurs. Un exemple typique d'application militaire est le déploiement des capteurs dans les endroits stratégique ou difficile d'accès, afin de surveiller toutes les activités des forces ennemies ou d'analyser le terrain avant d'y envoyer des troupes. Les RCSFs ont contribué dans le commandement, le contrôle, la communication, la surveillance du champ de bataille, la reconnaissance des forces d'opposition, l'évaluation des dommages de la bataille et la détection ainsi que la reconnaissance d'attaque à savoir si elle est nucléaire, biologique et chimique. La maîtrise de ces facteurs constitue un point fort et un atout pour les forces militaires [Akyildiz et al, 2002].
- **Environnementales**: les réseaux de capteurs peuvent être utilisés pour surveiller les changements environnementaux [CHALLAL, 2008]. comme, nous allons le voir ci-dessous :
 - La détection des feux de forêts.
 - La détection d'inondations et de tremblements de terre.
 - Le contrôle de l'environnement marin.
 - Les études de pollution.
 - La recherche météorologique ou géophysique.
 - Agriculture de précision : On trouve aussi l'utilité de ces réseaux dans la surveillance du niveau de pesticides dans l'eau potable, le niveau d'érosion du sol et le niveau de pollution atmosphérique.
- **Sanitaires** : Les RCSFs sont récemment très répandus dans le domaine de la médecine. Dans certains hôpitaux, ils ont installé des capteurs pour surveiller des données physiologiques des patients et pour surveiller aussi le comportement des médecins et des patients à l'intérieur d'un hôpital [Akyildiz et al, 2002]. D'autres

applications ont été envisagées à savoir : la surveillance du niveau du glucose, le contrôle des organes, la détection du cancer, etc. L'idée d'intégrer des capteurs biomédicaux sans fil dans le corps humain est prometteuse, bien qu'ils existent beaucoup de défis et de contraintes à prendre au sérieux : le système doit être ultra prudent, sans danger sur la santé du patient et stable et, nécessite peu de maintenance. Avec beaucoup de recherches et de progressions dans ce domaine, un coût de traitement médical peut être réduit et une qualité de vie meilleure est promise.

- **Commerciales et industrielles:** Les RCSFs offrent une grande flexibilité d'emploi, tel que l'intégration des micros capteurs dans un processus de stockage et de livraison de marchandises. Ils peuvent être utilisés pour connaître la position, l'état et la direction d'un colis. Il devient alors possible pour un client qui attend la réception d'un colis, d'avoir un avis de livraison en temps réel et de connaître la localisation actuelle de ses marchandises. Pour les entreprises manufacturières, les réseaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré [Akyildiz et al, 2002] [Amundson et Koutsoukos, 2009].
- **Autres applications:** Les réseaux de capteurs peuvent également être utilisés pour: surveiller l'infrastructure, lutter contre le terrorisme, contrôler le trafic, détecter des intrusions, l'urbanisme, l'ingénierie civile (surveillance des structures, les capteurs peuvent être placés dans les ponts afin de détecter et de signaler les faiblesses structurelles, dans les réservoirs d'eau pour détecter les matières dangereuses), le recouvrement des catastrophes (par exemple chercher des signes de vie après un tremblement de terre), et beaucoup d'autres applications qui rendent notre entourage plus intelligent.

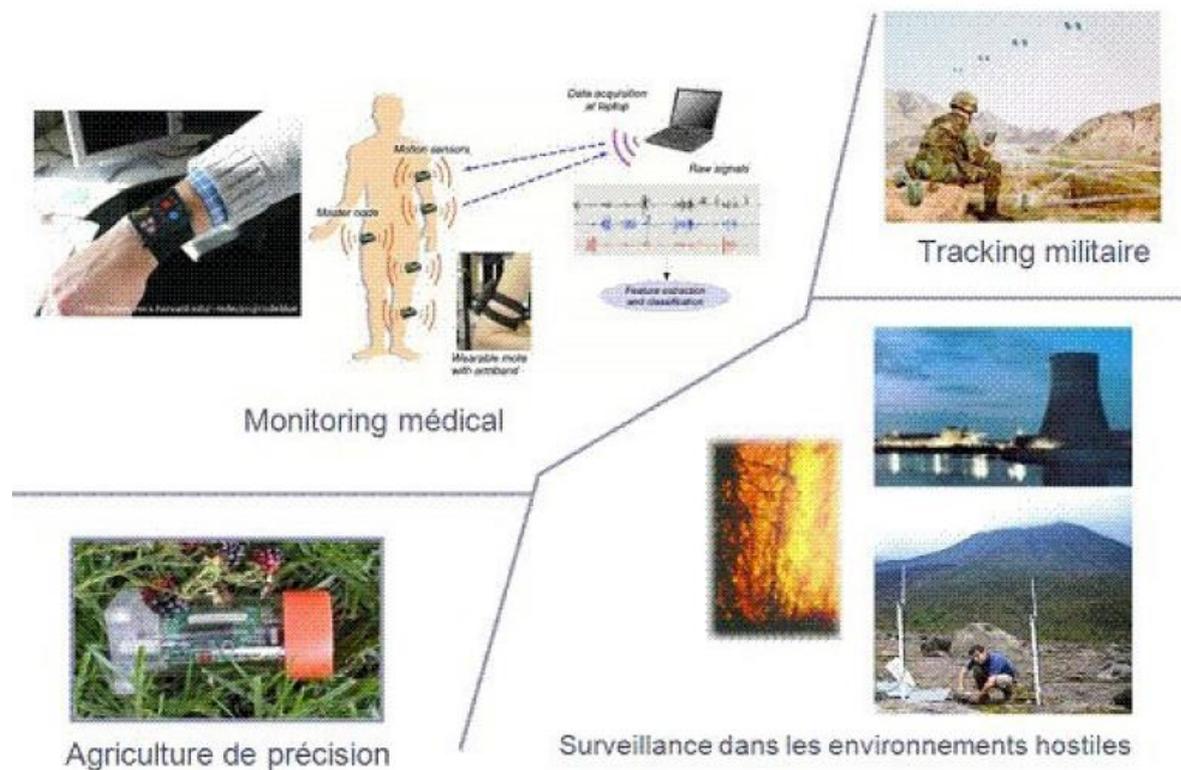


Figure I.5 : Quelques applications des RCSFs

I.5 Défis des réseaux de capteurs sans fil

Il y a plusieurs problèmes qui font face au progrès des RCSF. Parmi ceux-ci figurent les suivants [Deng et al, 2008] [Nicopolitidis et al, 2003] :

- (1) **Scalabilité.** La plupart des nœuds dans les réseaux de capteurs intelligents sont stationnaires. Des réseaux d'un grand nombre de nœuds de l'ordre de 10 000 ou plus sont attendus. Cela signifie que la mise à l'échelle est un problème crucial dans la conception ou le lancement de tout nouveau RCSF car nous aimons voir une amélioration proportionnelle des performances à mesure que la taille du réseau augmente. Les algorithmes et les protocoles conçus pour les RCSF doivent tenir compte des coûts de communication par rapport à la taille du réseau.
- (2) **Puissance limitée.** Étant donné que les RCSF sont souvent installés dans des zones reculées telles que les déserts, les forêts ou les zones militaires, leurs nœuds sont généralement alimentés par des batteries à durée de vie limitée. La recharge de telles batteries peut ne pas être possible. Compte tenu de cette contrainte, la durée de vie d'un nœud est déterminée par la durée de vie de la batterie qui l'alimente. En conséquence, la réduction de la puissance consommée est vitale. Il existe des

protocoles et des schémas qui ont été proposés pour contrôler la consommation d'énergie des RCSFs. Ces schémas sont basés sur des protocoles MAC éco-énergétiques, l'agrégation des données, la gestion de la topologie, la compression des données ou l'utilisation intelligente des batteries. Bien sûr, l'utilisation d'appareils électroniques et de puces qui consomment moins d'énergie est également un problème de conception clé.

- (3) **Auto-organisation.** Étant donné que les RCSF peuvent être installés dans des environnements hostiles, il est essentiel qu'ils soient conçus pour être auto-organisés. Les nœuds peuvent échouer en raison d'un environnement difficile ou de l'épuisement des batteries; par conséquent, le réseau doit pouvoir se reconfigurer périodiquement afin de pouvoir continuer à fonctionner et de nouveaux nœuds peuvent être ajoutés, si possible. Les nœuds individuels peuvent être déconnectés du réseau, mais la majeure partie du réseau doit continuer à fonctionner.
- (4) **Objectif final.** L'objectif ultime d'un RCSF n'est pas seulement la communication. Il doit détecter et estimer certains événements intéressants. Afin d'améliorer les capacités de détection et d'estimation, il est utile de fusionner les données de plusieurs capteurs. Une telle fusion de données nécessite la transmission de données et de messages de contrôle, ce qui peut limiter la conception et la structure du réseau. En outre, il est essentiel de faire la distinction entre les fausses données recueillies et les données reflétant une véritable urgence. Par exemple, une température élevée en usine peut indiquer un véritable incendie ou peut être due à des erreurs de détection ou de traitement.
- (5) **La localisation.** Quand un terminal de RCSF (station de base) reçoit une information d'événement à partir d'un nœud capteur, cette information n'est considérée utile que lorsqu'on connaît l'emplacement ou la position exacte ou l'événement s'est produit. La localisation dans les RCSFs est l'un des points les plus importants qui régissent les RCSFs.
- (6) **Capacité d'interrogation.** Dans les environnements des RCSF, un utilisateur peut être amené à interroger un nœud individuel ou un cluster de nœuds sélectionné, pour la collecte d'informations dans la zone. En fonction du degré de fusion des données effectué, il peut ne pas être possible d'envoyer un énorme volume de données sur le réseau. En variante, différents nœuds récepteurs voisins peuvent collecter des données d'une zone donnée et générer des messages récapitulatifs.

Une demande peut être envoyée au nœud récepteur le plus proche de l'emplacement préféré.

- (7) **Interopérabilité.** Avec les progrès impressionnants de la technologie de détection et de communication, nous commençons à voir des radios à courte portée bon marché, ainsi que des appareils et des liaisons réseau sans fil. Bien sûr, il est prévu que les RCSF seront largement déployés pour toutes sortes d'applications. Chaque nœud du réseau peut être équipé avec différents capteurs, y compris sismique, acoustique, caméra vidéo et lumière infrarouge, entre autres. Les nœuds peuvent être configurés en groupes et ils peuvent se synchroniser les uns avec les autres de manière à ce que les événements qui se produisent localement soient identifiés par la majorité, sinon la totalité, des nœuds du cluster. Ces nœuds collaboreront afin de prendre des décisions locales sur la base des données collectées par chaque nœud du cluster. Dans un tel agencement, un nœud peut agir comme nœud maître et les autres peuvent agir comme esclaves.
- (8) **Coût.** Un problème important dans le coût des microcontrôleurs sans fil est la taille de la mémoire nécessaire. Les concepteurs des réseaux de capteurs sans fil s'attendent à avoir accès à une gamme de puces ou de microcontrôleurs sans fil avec une taille de mémoire optimisée pour répondre aux besoins d'une variété d'applications. De même, la nécessité de développer des applications plus importantes telles que des dispositifs de passerelle et de développer une couche réseau tierce, montre qu'il existe un besoin pour une taille de mémoire beaucoup plus grande, supérieure à 250 ko dans certains cas.
- (9) **Temps de transmission.** Un problème qui est parfois négligé est le temps nécessaire pour envoyer les paquets. Le temps de transmission affecte les performances, la qualité de service, la consommation d'énergie et les interférences. Il est nécessaire d'avoir une transmission de données fiable et une durée de vie de la batterie prolongée dans les réseaux de capteurs sans fil. Nous pouvons améliorer la fiabilité de la transmission des données en utilisant une petite taille de paquet pratique, car cela donne la plus forte probabilité qu'un paquet soit livré à la destination en présence d'interférences. La durée de vie prolongée de la batterie est obtenue en minimisant la durée de fonctionnement de l'appareil radio, où la majeure partie de l'énergie est consommée. En général, une petite taille de paquet et une transmission occasionnelle peuvent aider à atteindre cet objectif d'économie d'énergie.

- (10) **Compression des données.** La compression des données du capteur avant la transmission peut réduire considérablement le temps de transmission. Dans les nœuds capteurs tels que les capteurs de niveau de gaz, de température, de pression et de niveau de lumière, la transmission de données sur la transition ou l'exception, au lieu des transmissions planifiées normales, est un moyen efficace de minimiser le trafic réseau. De plus, avoir la capacité d'effectuer un filtrage numérique ou une compression des données au niveau du nœud capteur est une approche intéressante pour minimiser la taille des données ainsi que le taux de récurrence des transmissions.
- (11) **Interférence et environnement.** En général, les interférences provenant d'autres réseaux sans fil à proximité tels que Bluetooth ou LAN sans fil doivent être traitées. Habituellement, cela ne présente qu'un état transitoire d'interférence avec le RCSF. Par exemple, la capacité d'un réseau IEEE 802.15.4 ou ZigBee à effectuer une répétition automatique dépassera probablement tout effet des interférences de Bluetooth. De même, pour les RCSF utilisant des transmissions occasionnelles et pour Bluetooth avec saut de fréquence, la probabilité d'une collision de trame est faible. En utilisant des schémas d'évitement des collisions, les réseaux locaux sans fil (WLAN) peuvent écouter un canal radiofréquence (RF) clair avant d'envoyer des données. Cependant, dans des conditions de trafic intense dans les WLAN, nous pouvons obtenir une disponibilité limitée du canal RF vers le RCSF en raison de l'état continu d'interférence. Dans une telle situation, il est recommandé de définir le RCSF sur un canal différent. Les structures de bâtiments environnantes affectent également l'environnement RF. Sols en béton armé, acier, pierre, les murs et les ressources de construction analogues apportent des niveaux d'atténuation élevés ainsi que des évanouissements par trajets multiples. De même, le mouvement de personnes ou d'équipements affecte considérablement le niveau du signal à n'importe quelle position spécifique. En général, les effets des structures de bâtiment complexes peuvent être atténués en utilisant des nœuds de routeur supplémentaires dans un réseau maillé qui sont installés pour contourner ces obstacles.
- (12) **Sécurité.** En raison des caractéristiques du support de communication sans fil, il existe divers défis de sécurité auxquels sont confrontés les RCSFs, notamment l'écoute clandestine, l'attaque de l'homme du milieu, l'usurpation d'identité et le déni de service distribué (DDoS). L'inquiétude pour la sécurité dans les RCSFs peut

être encore plus grande que celle dans un réseau sans fil Adhoc traditionnel car, dans de nombreux cas, les limitations de calcul et de consommation d'énergie créent des obstacles à la mise en œuvre de solutions de sécurité puissantes et efficaces dans les RCSF. Par conséquent, les avancées dans la conception de mécanismes de sécurité dans les réseaux RCSF pour protéger la confidentialité, la disponibilité et l'intégrité sont essentielles au bon fonctionnement de ces systèmes [Nicolitidis et al, 2003] [Obaidat et al, 2009]. L'acceptation de la norme AES (Advanced Encryption Standard) avec une longueur de clé de 128 bits garantit l'intégrité des données et la résistance au piratage. Un schéma de sécurité AES peut être implémenté dans le logiciel, tandis qu'un processeur de chiffrement matériel dédié offre une meilleure solution car cela réduit les frais généraux du logiciel et permet une opération de chiffrement / déchiffrement plus rapide. De toute évidence, cela est essentiel pour les nœuds capteurs, qui doivent passer le moins de temps possible à s'éveiller, car rester éveillé consomme beaucoup d'énergie de la batterie du nœud. De plus, rendre la puce de chiffrement AES accessible au logiciel d'application facilite un niveau de sécurité plus élevé [Obaidat et al, 2009].

I.6 Conclusion

Les progrès connus récemment dans le domaine des technologies sans fil, ont permis l'apparition des réseaux de capteurs. Les propriétés de ce type de réseaux en termes de flexibilité, de coût réduit, et de facilité de déploiement offrent de nombreuses possibilités de développement dans divers domaines. Pour mettre en place un réseau de capteurs le déploiement des nœuds est la première opération dans leur cycle de vie. Selon les besoins des applications, on peut envisager deux formes de déploiements, manuellement (déterministe) les un après les autres ou bien aléatoirement (indéterministe) par avion, par exemple. Dans un grand nombre d'applications des RCSFs, le déploiement manuel des capteurs est impossible. De plus, même lorsque l'application permet un déploiement déterministe, le déploiement aléatoire est adopté dans la majorité des scénarios à cause des raisons pratiques tels que le coût et le temps.

Ce qui déclenche le problème de la localisation dans ce type de réseaux. Ce chapitre a été axé sur la présentation des concepts fondamentaux des RCSFs. Le chapitre suivant décrira le concept de la localisation dans les réseaux de capteurs sans fil.

Chapitre II

Problème de localisation dans les RCSFs

Chapitre II

Problème de localisation dans les RCSFs

II.1. Introduction

Dans la majorité des applications des réseaux de capteurs sans fil, un événement détecté n'est utile que si une information relative à sa localité géographique est fournie. C'est le cas de la surveillance des feux de forêt ou de troupes d'ennemies dans un contexte militaire. Sans cette information, ces applications n'auraient aucun sens. Il s'agit donc de déterminer pour chacun des capteurs sa position.

La localisation deviennent plus qu'indispensable, non seulement, à l'exploitation des données récoltées, mais également, au fonctionnement du réseau (routage géographique, couverture de la zone d'intérêt, agrégation des données, etc.). Les systèmes de localisation automatique existants sont nombreux et utilisent plusieurs voies technologiques. Ils sont très différents les uns des autres et répondent en général à des besoins différents (localisation dans une pièce ou à l'échelle de la planète, localisation précise ou imprécise, localisation d'un seul ou plusieurs objets, localisation statique ou suivi de cible, . . .).

Nous présentons dans ce qui suit le problème de la localisation statique dans les réseaux de capteurs, leurs caractéristiques et contraintes, les différentes méthodes pour l'estimation des distances entre nœuds, et nous finalisons par un aperçu des algorithmes de localisation existants.

II.2. Définition de la localisation

Le mot localisation en dictionnaire signifie l'action de déterminer l'emplacement où se situe une chose, un phénomène ou son origine. La localisation dans les réseaux de capteurs est un procédé permettant d'identifier une position de différents capteurs déployés de manière aléatoire sur un plan ou une carte géographique. Cette opération est réalisée à l'aide d'un terminal capable d'être localisé en temps réel ou de façon différée [Kulaib et al, 2011].

II.3. Système de localisation

II.3.1 Critères d'un système de localisation

Un système de localisation doit respecter un ensemble de contraintes imposées par les spécificités des RCSF dû à leurs limitations (ressources matérielles : puissance du processeur, mémoire, énergie) et applications. Il est très intéressant de les découvrir même si dans la pratique, il est impossible de tenir compte de tous ces critères lors du développement d'un algorithme de localisation. Parmi ces contraintes [Boukerche, 2008] [KARA, 2009] :

1. **Précision de la localisation** : Nous parlons de l'erreur (ou la distance euclidienne) qu'il y a entre les vraies positions des nœuds et les positions estimées par la méthode de localisation.
2. **Evolutivité (extensibilité, passage à l'échelle)** : Est-ce qu'un algorithme fonctionne sur un réseau de plusieurs milliers de nœuds ? Et si oui, est-il toujours aussi efficace. Ce critère est en rapport avec le fait qu'un algorithme soit implémenté de façon distribuée ou centralisée. La plupart des algorithmes de localisation sont sensibles à la densité des nœuds ; s'il n'y a pas assez de nœuds l'algorithme donne des résultats imprécis [Stojmenovic, 2005].
3. **Robustesse au bruit** : Il faut analyser comment un algorithme se comporte face au bruit rencontré dans les mesures de distances avec les voisins.
4. **Réactivité du système** : Avec quelle rapidité le système de localisation nous renvoie-t-il les positions des nœuds ? Ceci est particulièrement important lorsque l'on veut s'occuper des nœuds mobiles et de suivi de cibles.
5. **Tolérance à la basse connectivité** : Est-ce qu'un algorithme fonctionne dans un réseau à basse connectivité (un réseau où chaque nœud ne sait communiquer qu'avec un petit nombre de ses voisins) ? Comment sont affectées les performances d'un algorithme face à cette situation ?
6. **Efficacité dans l'utilisation des ressources du RCSF (Coût énergétique)** : Même si le système de localisation est indispensable pour le bon fonctionnement du RCSF, il n'est pas le but principal du réseau. Il doit utiliser les ressources du capteur d'une manière efficace pour ne pas les épuiser. Comme le facteur dominant de la consommation d'énergie est la communication radio, il faut trouver un

algorithme qui communique le moins possible via la radio.

II.3.2 Caractérisations d'un système de localisation

Il existe de nombreuses approches pour résoudre le problème de la localisation et chaque méthode a ses avantages et ses inconvénients. Un ensemble de propriétés permettent de comparer les systèmes de localisation [Boukerche, 2008] [KARA, 2009] [Stojmenovic, 2005] [Karl et Willig, 2005] [Hightower et, Borriello 2001]. Les plus importantes propriétés sont les suivantes :

II.3.2.1 Système des cordonnées

Un repère qui permet d'obtenir des positions et qui les organise de façon cohérente. Trois types de position sont observés:

1. **Physiques (absolues) :** Cette propriété définit le fait que le système de localisation fournit aux nœuds des positions réelles (valeurs numériques) sur le globe terrestre (longitude et latitude) ou dans l'espace (longitude, latitude et altitude).
2. **Relatives :** Indiquent juste une direction par rapport à un voisinage donné à droite, à gauche, au bout de la rue par exemple.
3. **Symboliques :** désignent par exemple une salle, un espace particulier. comme une personne se trouve dans la cuisine, au salon, etc.

II.3.2.2 Forme d'implémentation

Cette caractéristique fait référence à l'endroit où les positions sont calculées. Nous distinguons deux façons d'implémenter un algorithme de localisation [Girod et al, 2002] :

1. **Centralisé :** les nœuds communiquent avec leurs voisins et renvoient les mesures vers une station de base centrale qui calcule les positions et les renvoie vers les nœuds concernés.
2. **Distribué :** le calcul de la position se fait localement par le nœud en se basant sur des mesures effectuées localement.

En outre les aspects de mise à l'échelle (extensibilité) et d'efficacité (coût de communication et de calcul en terme énergétique), le respect de la vie privée est très

important, il n'est pas toujours souhaitable (surtout pour les grands réseaux) qu'un nœud révèle sa position à un serveur central.

II.3.2.3 Utilisation d'estimations de distances

Selon cette caractéristique, nous distinguons deux types d'algorithmes de localisation [He et al, 2003] :

1. **Range-free** : Ces méthodes ne calculent pas de distances entre voisins. Elles utilisent la connectivité pour identifier la position des nœuds.
2. **Range-based** : Ces méthodes estiment les distances entre les nœuds et ensuite dérivent de ces distances les positions des différents nœuds du réseau.

II.3.2.4 Nécessité des ancres

L'ancre (ou balise) est un capteur qui est au courant de son emplacement, que ce soit par GPS ou nous le mettons manuellement lors du déploiement des nœuds du RCSF [Hightower et, Borriello 2001]. Donc, le fait que la méthode de localisation requiert ou non de connaître la position d'un certain nombre d'ancres est une caractéristique importante de la méthode.

1. **Les méthodes anchor-based** : Sont celles qui ne fonctionnent pas sans connaître la position d'un certain nombre d'ancres à priori. Ici, nous supposons que nous connaissons la position de quelques nœuds du réseau dans un certain système de coordonnées, les positions des autres nœuds dans ce système de coordonnées peuvent être trouvées.
2. **Les méthodes anchor-free** : Sont celles qui n'ont besoin de la position d'aucun nœud pour fonctionner ; elles créent donc une carte relative du réseau. Ici, un ensemble de transformations géométriques (rotation, translation et éventuellement homothétie) sont nécessaires pour obtenir les coordonnées des nœuds.

II.3.3 Phases principale d'un système de localisation

Les méthodes de localisation ont pour but d'estimer les positions des différents nœuds inconnu de manière automatique. Dans le cas général, une méthode de localisation dans

les réseaux de capteurs peut être décomposée en trois phases distinctes [Boukerche, 2008] [KARA, 2009]. Chaque phase a ces propres techniques de résolution, elles seront détaillées dans la suite de ce chapitre. Ces trois grands phases ou sous-systèmes sont [Paul et Sato, 2017] :

1. **L'estimation de distance** : Dans cette phase, les nœuds communiquent entre eux et collectent différents indicateurs de qualité des communications radios. à partir desquelles les distances séparant les nœuds peuvent être estimées. Cette information sur la distance entre nœuds est utilisée par la suite par les deux autres phases (estimation locale).
2. **Dérivation des positions** : Cette phase permet de trouver les positions des nœuds qui respectent au mieux les distances et/ou d'angles inter-nœuds estimées et sur les positions des ancres disponibles en utilisant les techniques de la géométrie (triangulation par exemple).
3. **Algorithme de localisation** : Elle définit la manière avec laquelle les informations disponibles (ancres, distances, angles, positions des nœuds déjà estimées) sont manipulées afin que la grande partie ou la totalité des autres nœuds inconnue permettent d'estimer leurs positions.

La performance de chaque sous système (phase) influencera directement sur la performance total du système de localisation. La Figure II-1 illustre les différentes phases d'un système de localisation. Dans le cadre de système de localisation, les cercles gris représentent les ancres et les cercles en blanc représentent des nœuds inconnus (celles qui ne sont pas encore localisés).

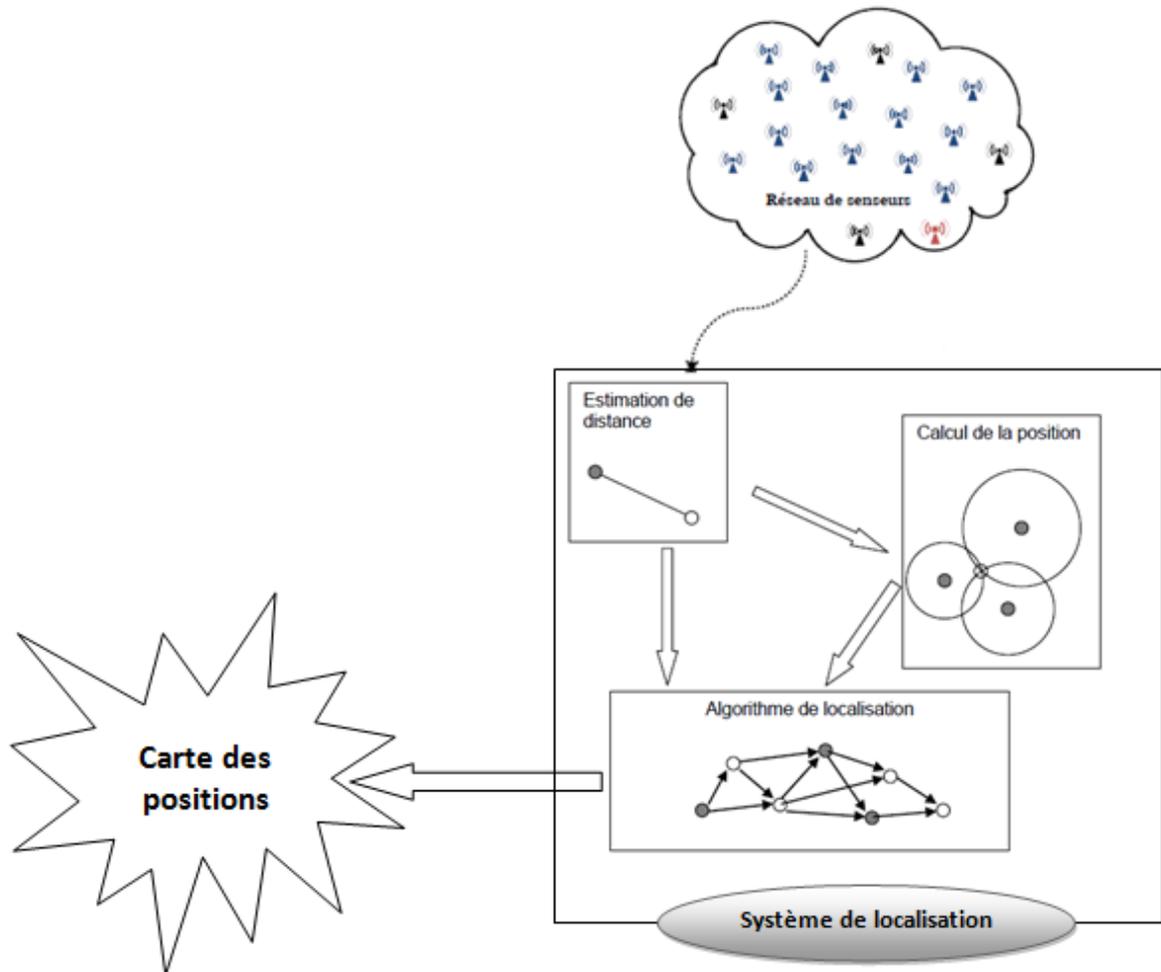


Figure II-1. Phases d'un système de localisation

II.4. Modèle d'estimation des distances

Cette phase d'estimation est très importante car elle sera utilisée par la suite pour les deux autres phases du système, le calcul de la position et l'algorithme de localisation. Plusieurs modèles existants pour estimer la distance/angle. Certains sont précis mais coûteux, nécessite un matériel supplémentaire, énergie, et des ressources processeur. D'autres sont imprécis mais ils sont déjà disponibles sur la plupart des capteurs et qui ne nécessitent aucun matériel supplémentaire [KARA, 2009].

L'estimation de la distance peut se faire sur la base de différents indicateurs de mesure parmi ces, nous distinguons:

- ✓ **RSSI** (Received Signal Strength Indicator) : La puissance du signal à la réception.

- ✓ **LQI** (Link Quality Indicator) : Le taux d'erreurs corrigées lors des transmissions.
- ✓ **ToA/TDoA** : temps de parcours.
- ✓ **AoA/ DoA** (Angle of Arrival / Direction of Arrival): angle d'arrivée et la portée de communication (nombres de sauts entre nœuds).

Un exemple intuitif d'estimation des distances est donné dans la Figure II.2. Dans cet exemple, l'algorithme de localisation requiert toutes les distances :

1. Les valeurs de l'indicateur pour les liens de communication sont collectées.
2. Ces valeurs sont transformées en distances.
3. Les distances manquantes sont complétées.

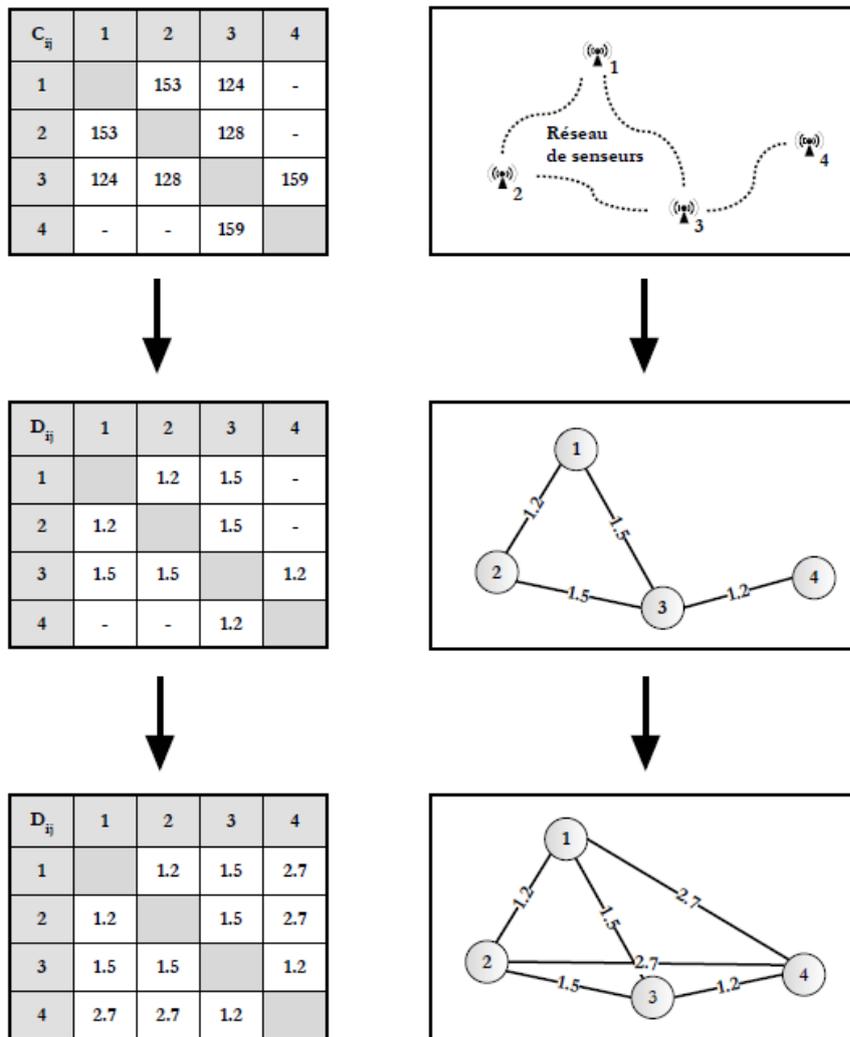


Figure II.2 Exemple de collecte d'informations de voisinage.

II.4.1 Le RSSI (Received Signal Strength Indicator)

C'est un indicateur de la puissance de signal au moment de la réception d'un message. Cette technologie considère la perte de puissance d'un signal entre son émission et sa réception. Cette perte (path loss, c'est la diminution de puissance du signal radio) varie en fonction de la distance entre les deux capteurs : plus les capteurs sont éloignés, plus la perte est importante. Cette perte sera alors traduite en une distance [Cho et al, 2011].

La valeur du RSSI est une valeur numérique sur 8 bits stockée dans un registre du hardware de la radio. Un modèle très répandu existe pour représenter le RSSI en fonction de la distance :

$$RSSI(d) = P_T - PL(d_0) - 10\eta \log_{10} \frac{d}{d_0} + X_\delta \quad (I, 1)$$

Où :

- P_T est la puissance de transmission.
- $PL(d_0)$ est la perte de signal pour une distance de référence d_0 .
- η est l'exposant de la perte de signal.
- Toutes les puissances sont données en dBm et toutes les distances en mètres.

Les variations aléatoires dans les RSSI sont exprimées comme une variable aléatoire gaussienne de moyenne 0 et de variance. $\delta^2: X_\delta \sim N(0, \delta^2)$

D'autres modèles plus complexes existent [Bahl et Padmanabhan, 2000] [Cho et al, 2011]. En général, chaque modèle est conçu pour un environnement et un scénario bien défini. Le grand avantage de la méthode est son faible coût car tous les nœuds sont déjà équipés de dispositif de communication sans fil ; de plus, elle ne rajoute aucun surcoût de communication. Son inconvénient est sa sensibilité aux bruits.

II.4.2 Le LQI (Link Quality Indicator)

Le LQI est une indication de la puissance et/ou de la qualité d'un paquet reçu. Cette valeur fournit une indication sur le taux d'erreurs à la réception d'un message. Il s'agit de la moyenne des erreurs sur les 8 premiers symboles reçus après le SFD (Start of Frame Delimiter : une séquence de bits qui indique le début d'un message). La valeur est comprise entre 50 et 110. Une valeur de 110 correspondant à une transmission sans erreur et une valeur de 50 à un taux critique d'erreurs (proche de la corruption et du rejet du paquet).

II.4.3 ToA/ToF (Time of Arrival, Time of Flight)

La technique ToA exploite la relation entre la distance parcourue par un signal et le temps de parcours. Dans ToA, la distance entre un capteur émetteur et un autre récepteur est la multiplication du temps pris par le signal avec la vitesse des ondes radio.

$$\text{Distance} = \text{Temps} * \text{Vitesse}$$

La vitesse de propagation du signal radio est égale 3×10^8 m/s.

Selon le type du signal utilisé, la méthode ToA nécessite, en général, des horloges à très grande résolution du temps pour obtenir des précisions acceptables [Boukerche, 2008]. Cette technologie est celle utilisée par le système GPS.

Pour les ondes radio, la résolution du temps doit être très élevée. Cependant, pour les ondes sonores, les besoins en résolution du temps sont modestes, mais leur inconvénient est que leur vitesse de propagation dépende des conditions extérieures comme la température et l'humidité [Karl et Willig, 2005].

II.4.4 TDoA (Time Difference of Arrival)

Cette technologie s'applique dans les cas où, chaque capteur muni d'un haut-parleur et un microphone. Certains systèmes utilisent les ultrasons tandis que d'autres utilisent une fréquence audible [Niculescu et Nath, 2003].

Le TDoA se base sur la différence des temps d'arrivée d'un ou plusieurs signaux et suppose également que la vitesse de propagation des signaux est connue. En général, un Capteur A envoie un signal radio suivi d'un signal son. En déterminant la différence de temps entre l'arrivée des deux signaux, un Capteur B peut estimer sa distance à partir de A.

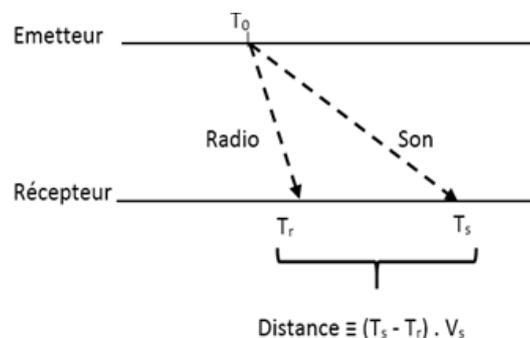


Figure II.3 Principe du TDoA

Avec ;

T_r : le temps d'arrivée du signal radio.

T_s : le temps d'arrivée du signal sonore.

V_s : la vitesse de signal sonore.

Cette technique est limitée par la distance à mesurer (inférieure à une centaine de mètres), car on suppose que le temps de propagation du signal radio est quasiment nul par rapport au temps de propagation du signal ultrasonore [Yu et Oppermann, 2004].

II.4.5 AoA/DoA (Time Difference of Arrival)

La technique AOA permet de mesurer l'angle formé entre deux capteurs au lieu de la distance. Chaque capteur est doté d'antennes orientées de sorte à déduire l'angle qu'il forme avec un voisin lorsque ce dernier lui envoie un signal. Cet angle est reporté par rapport à un axe propre au capteur. Toutefois, un capteur peut être équipé d'une boussole est dans ce cas, l'angle sera reporté sur un des axes nord, sud ou ouest [Cardei et al, 2005] [Boushaba et al, 2009]. La **Figure II.4** illustre le principe de la méthode AOA. Cette méthode nécessite d'avoir du matériel supplémentaire (des antennes ou récepteurs ultrasons).

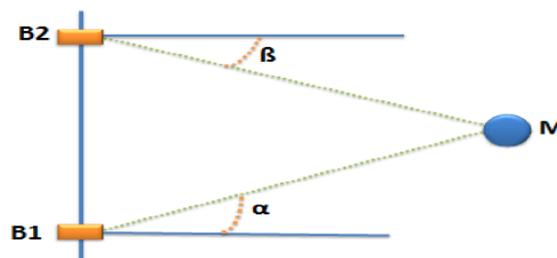


Figure II.4 Principe de la méthode AOA.

II.4.6 Bilan des techniques de mesures

Le Tableau 2-1 récapitule les caractéristiques essentielles des techniques d'estimation des distances décrites dans ce chapitre.

Tableau II-1 Comparaison des techniques d'estimation des distances [KARA, 2009].

Méthode	Précision	Distance Maximale	Matériel supplémentaire	Défis (challenges)
RSSI	2-4 m	La portée	Rien	Variations du RSSI, Interférences
TOA	2-3 cm	La portée	Rien	Synchronisation des nœuds
TDOA	2-3 cm	Quelques mètres (2-10 m)	Capteur à ultrason	La distance maximale
AOA	Quelques degrés (5°)	La portée	Un ensemble de récepteurs	Travailler avec des nœuds de petite taille

II.5. Mécanismes de dérivation des positions

Une fois qu'un nœud aurait suffisamment d'information de distances/angles entre lui et certain nombre des nœuds références (ancres), ainsi que les positions de ces ancres, il peut calculer sa position. Pour cela, plusieurs méthodes géométriques sont utilisées. Parmi ces méthodes, nous retrouvons la trilatération, la multilatération et la triangulation.

Le choix entre ces méthodes de dérivation de la position influe sur les performances finales du système de localisation. Ce choix dépend des informations disponibles et des ressources du processeur (CPU, mémoire, etc.). Dans les suivants paragraphes nous présentons les méthodes géométriques de calcul de la position, suivies par un tableau de comparaison de leurs caractéristiques.

II.5.1 Multilatération

La multilatération est la méthode la plus basique et intuitive. Cette méthode calcule la position d'un nœud par l'intersection de cercles. Pour estimer sa position, un nœud a besoin de connaître la position d'un certain nombre d'ancres (au moins trois dans un espace 2D) ainsi que sa distance par rapport à ces ancres. La distance peut être calculée en utilisant l'une des méthodes décrites dans la partie précédente «modèle d'estimation de distance».

Soit une cible 'a' dont on veut trouver la position x_a , et soit 'm' ancres i dont nous connaissons les positions $x_i \in R^p, 1 \leq i \leq m$

Nous supposons que nous connaissons aussi une estimation des distances $\hat{d}_i, 1 \leq i \leq m$ entre chaque ancre i et le **nœud** a . Nous pouvons alors poser :

$$\begin{aligned} (x_{11} - x_{a1})^2 + (x_{12} - x_{a2})^2 + \dots + (x_{1p} - x_{ap})^2 &= \hat{d}_{1a}^2 \\ &\vdots \\ (x_{m1} - x_{a1})^2 + (x_{m2} - x_{a2})^2 + \dots + (x_{mp} - x_{ap})^2 &= \hat{d}_{ma}^2 \end{aligned} \quad (II, 2)$$

Le système peut être linéarisé en soustrayant la dernière équation des $m-1$ équations précédentes.

$$\begin{aligned} x_{11}^2 - x_{m1}^2 - 2(x_{11} - x_{m1})x_{a1} \\ + x_{12}^2 - x_{m2}^2 - 2(x_{12} - x_{m2})x_{a2} \\ + \dots \\ + x_{1p}^2 - x_{mp}^2 - 2(x_{1p} - x_{mp})x_{ap} \\ &\vdots \\ x_{(m-1)1}^2 - x_{m1}^2 - 2(x_{(m-1)1} - x_{m1})x_{a1} \\ + x_{(m-1)2}^2 - x_{m2}^2 - 2(x_{(m-1)2} - x_{m2})x_{a2} \\ + \dots \\ + x_{(m-1)p}^2 - x_{mp}^2 - 2(x_{(m-1)p} - x_{mp})x_{ap} \end{aligned} = \begin{aligned} &\hat{d}_{1a}^2 - \hat{d}_{ma}^2 \\ &\vdots \\ &\hat{d}_{(m-1)a}^2 - \hat{d}_{ma}^2 \end{aligned} \quad (II, 3)$$

En réordonnant les termes, nous obtenons un système d'équations linéaires de la forme $Ax=b$ où :

$$A = \begin{bmatrix} 2(x_{11} - x_{m1}) & \dots & 2(x_{1p} - x_{mp}) \\ \vdots & \ddots & \vdots \\ 2(x_{(m-1)1} - x_{m1}) & \dots & 2(x_{(m-1)p} - x_{mp}) \end{bmatrix} \quad (II, 4)$$

$$b = \begin{bmatrix} x_{11}^2 - x_{m1}^2 + \dots + x_{1p}^2 - x_{mp}^2 + \hat{d}_{ma}^2 - \hat{d}_{1a}^2 \\ \vdots \\ x_{(m-1)1}^2 - x_{m1}^2 + \dots + x_{(m-1)p}^2 - x_{mp}^2 + \hat{d}_{ma}^2 - \hat{d}_{(m-1)a}^2 \end{bmatrix} \quad (II, 5)$$

Comme nous avons des erreurs dans les estimations de distances, nous ne pouvons pas trouver de solution exacte à ce système d'équations. La solution la plus proche (au sens des moindres carrés) de la solution exacte est obtenue en minimisant $(Ax-b)^T(Ax-b)$ [Bontempi, 2006].

Et donc $\hat{x}_a = \operatorname{argmin}_x (Ax - b)^T (Ax - b) = (A^T A)^{-1} A^T b$. Où $\hat{X}_a = \begin{bmatrix} \hat{x}_{a1} \\ \hat{x}_{a2} \\ \vdots \\ \hat{x}_{ap} \end{bmatrix}$, c'est-à-dire notre

estimation de la position du nœud a .

II.5.2 Trilatération

Ceci est un cas particulier de la multilatération, Quand nous avons seulement l'information de distances et positions de trois ancrés [Zhong, 2008]. La trilatération calcule la position d'un nœud par l'intersection de trois cercles centrés aux positions des ancrés disponibles et qui ont comme diamètres la distance entre le nœud cible et chacun de ces ancrés, comme le montre la **Figure II-5**.

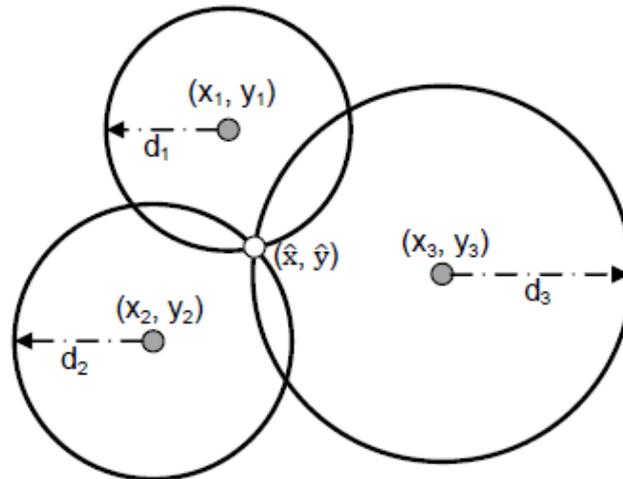


Figure II-5 Principe de la trilatération

II.5.3 Triangulation

La triangulation utilise les angles à la place des distances. La position de nœud cible est calculée en utilisant les lois de la trigonométrie (cosinus et sinus). Dans les RCSF, le plus important est que chaque nœud calcule sa position lui-même (auto-localisation). Dans ce cas, au moins trois ancrés sont nécessaires. Le nœud non localisé estime ses angles avec chaque ancre. En se basant sur les angles estimés et les positions des ancrés (qui forment un triangle), le nœud calcule sa position en utilisant les lois de la trigonométrie. La **Figure II-6(a)** illustre le principe de l'auto-localisation en utilisant la triangulation. Cette technique est similaire à la trilatération. En effet, en se basant sur les angles d'arrivée, il est possible de déduire les distances vers les divers ancrés [Niculescu et Nath, 2003].

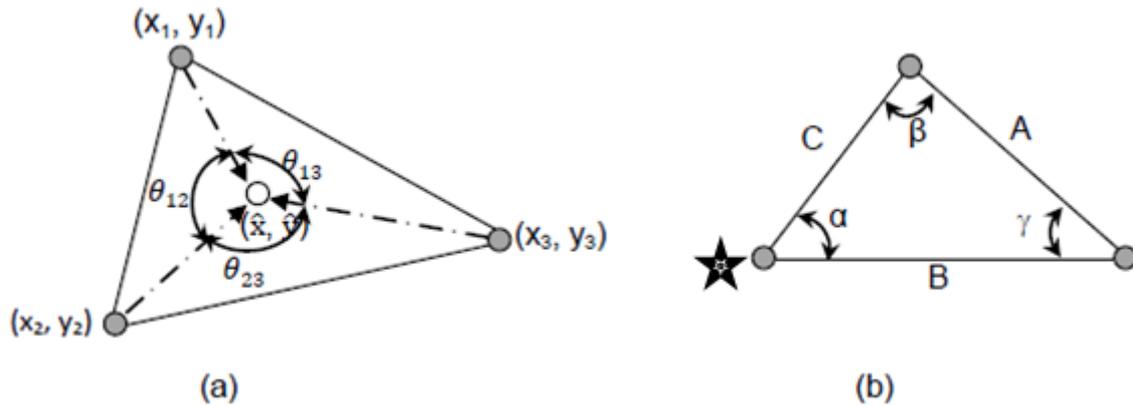


Figure II-6 Principe de la triangulation [Cardei et al, 2005]

Soit un triangle ABC, dans lequel on utilise les notations usuelles exposées sur la **Figure II-6(b)** : d'une part α , β et γ pour les angles et, d'autre part, a , b et c pour les cotés respectivement opposés à ces angles. Alors, les formules du sinus et cosinus respectivement stipule que :

$$\frac{A}{\sin\alpha} = \frac{B}{\sin\beta} = \frac{C}{\sin\gamma} \quad (II, 6)$$

$$\begin{aligned} c^2 &= A^2 + B^2 - 2AB\cos(\gamma) \\ B^2 &= A^2 + C^2 - 2AC\cos(\beta) \\ A^2 &= B^2 + C^2 - 2BC\cos(\alpha) \end{aligned} \quad (II, 7)$$

À partir de ces relations de deux nœuds de référence, on peut obtenir la position d'un troisième nœud (l'étoile par exemple **Figure II-6(b)**). Dans ce cas, on utilise en effet, la distance AC entre les deux ancres puis on mesure les angles A et C. Ensuite, il existe plusieurs solutions pour arriver à déterminer les deux distances AB et BC qui nous manquent.

II.5.4 Bilan sur les méthodes de dérivation de positions

Les trois méthodes qui peuvent être utilisées pour le calcul de la position ont été exposées. Il est à noter que cette présentation n'est pas exhaustive. Le choix de la méthode dépend des informations collectées et des ressources du processeur. Le Tableau II-2 compare les principales caractéristiques des trois méthodes présentées dans les paragraphes précédents.

Tableau II-2 Comparaison des méthodes de dérivation de positions [KARA, 2009].

Méthode	Nombre de références	Distance	Angle	Complexité temporelle	Défis
Trilatération	3	Oui	Non	O(1)	Sensible aux imprécisions des distances
Multilatération	$n \geq 3$	Oui	Non	O(n ³)	Complexité du calcul
Triangulation	3	Non	Oui	O(1)	Besoin de matériel supplémentaire

II.6 Un aperçu des algorithmes de localisation existants

Dans cette section, nous donnons un aperçu non exhaustif des algorithmes utilisés pour la localisation dans les réseaux de capteurs sans fil.

II.6.1 Bounding Box (Cadre englobant)

L'algorithme BB est un algorithme assez simple. Il utilise les distances \hat{d}_{ij} pour contraindre les coordonnées inconnues du nœud cible i par rapport aux coordonnées connues de l'ancre j . Ces contraintes prennent la forme suivante :

$$\begin{aligned} x_{j1} - \hat{d}_{ij} < x_{i1} < x_{j1} + \hat{d}_{ij} \\ x_{j2} - \hat{d}_{ij} < x_{i2} < x_{j2} + \hat{d}_{ij} \end{aligned} \quad (II, 8)$$

L'algorithme calcule la position d'une cible comme étant au centre du rectangle formé par l'intersection des rectangles formés par toutes les contraintes [Simic et Sastry, 2003]. Cette intersection peut être calculée facilement en prenant le maximum des coordonnées inférieures et le minimum des coordonnées supérieures [Boukerche, 2008].

$$(\hat{x}, \hat{y}) = \left(\frac{\max(x_i - d_i) + \min(x_i + d_i)}{2}, \frac{\max(y_i - d_i) + \min(y_i + d_i)}{2} \right) \quad (II, 9)$$

La **Figure II-7** illustre le principe de l'algorithme de Bounding Box.

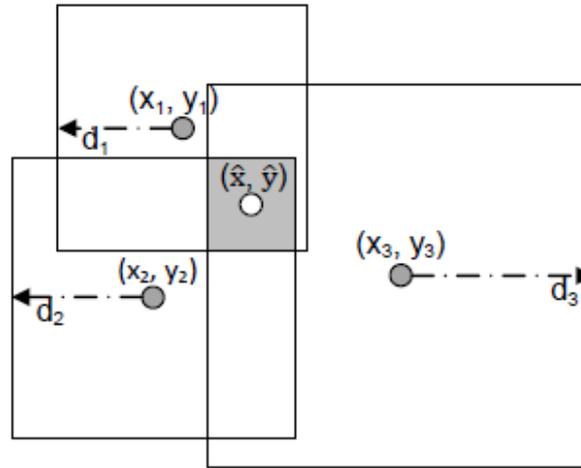


Figure II-7 Principe du cadre englobant

II.6.2 GPS-Less (centroïde)

Cette méthode présentée dans [Bulusu et al, 2000], se base essentiellement sur un certain nombre d'ancres qui ont des régions de couverture superposées et servent comme des points de référence. Pour nous permettre de localiser les différents nœuds inconnus, chaque ancre transmet un message 'beacon' tous les temps T , qui contient sa position. Un nœud voulant se localiser écoute pendant un certain temps les ancres. Il choisit un certain nombre d'ancres et estime sa position en prenant le centroïde des positions des ancres choisis selon l'équation II.10. :

$$posi_{estimé}(x, y) = \left[\frac{\sum_{i=1}^n x_i}{N} ; \frac{\sum_{i=1}^n y_i}{N} \right] \quad (II, 10)$$

Avec, $\sum_{i=1}^n x_i$, $\sum_{i=1}^n y_i$ la somme des x et y des ancres écoutés par le nœud à localiser respectivement, et N le nombre de ces ancres.

Pour choisir les ancres, il calcule pour chaque ancre i entendu un coefficient CM_i :

$$CM_i = \frac{\text{nombre de message envoyer par } i}{\text{nombre des messages reçus}} \times 100 \quad (II, 11)$$

Il calcule cela pendant un certain laps de temps (en sachant que tous les temps T chaque ancre est censé avoir émis un message) et choisit tous les nœuds qui sont au dessus d'un certain seuil (par exemple 95%).

Sur la Figure II.8, une réalisation de Centroid est illustrée avec uniquement les ancres retenues représentées par un point. La position estimée est représentée par un plus.

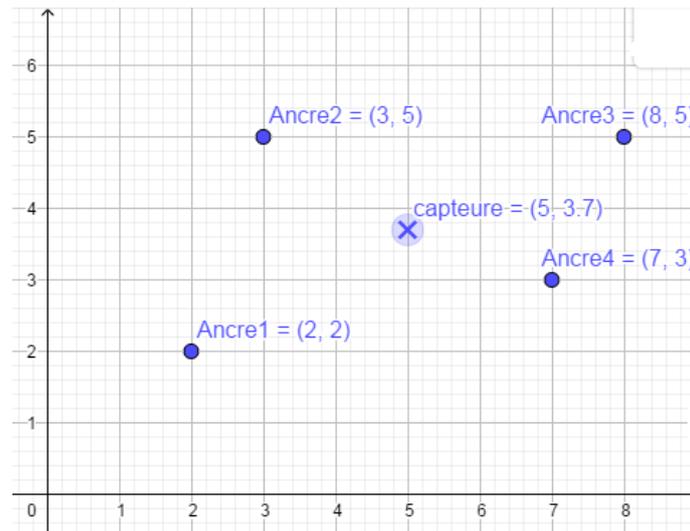


Figure II.8 : Exemple de localisation par centroïde.

L'inconvénient principal de cette méthode est le fait d'avoir à placer ces nœuds ancrés partout où l'on veut être localisable. De plus, les nœuds ancrés effectuent trop d'opérations de communication (émissions/réceptions) donc consomment une grande quantité d'énergie.

II.6.3 APIT

La méthode APIT [Huang et al, 2003] est spécialement optimisée pour le critère de la consommation d'énergie. Elle nécessite un certain nombre de nœuds ancrés comme pour le GPS-Less. Il est spécifié que ces nœuds ont des émetteurs longue portée. Ces ancrés émettent des messages contenant leur position en continu.

Un nœud qui veut se localiser écoute les ancrés passantes. Pour chaque triplet d'ancres différent découvert, il crée un triangle. Il rejette certains d'entre eux via un test appelé PIT test. Il trouve ensuite l'intersection des triangles restants et obtient ainsi une région dans laquelle se trouve le nœud. Le centroïde de cette région est pris comme estimation de la position. On peut voir cela de façon intuitive sur la Figure II.9.

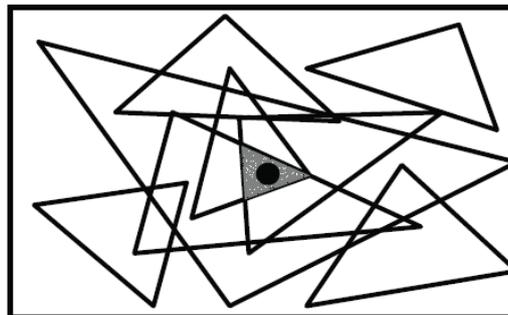


Figure II.9 Principe de la méthode APIT

II.6.4 MDS-MAP

MDS-MAP est basé sur une technique appelée la multidimensionnelle scaling (MDS) [Ji et Zha, 2004]. MDS est une méthode utilisée pour l'analyse des données de similarité ou non d'un ensemble d'objets. L'idée de MDS est simple, supposons que n points distribués dans un volume donné que nous ignorons leurs positions, mais nous connaissons les distances entre eux. Les informations sur les distances peuvent être représentées sous forme de carte dans un espace bidimensionnel ou tridimensionnel. En fait, le MDS-MAP s'appuie sur la technique MDS pour déterminer les positions relatives des nœuds. Il est divisé en trois étapes :

Etape 1: Dans un premier temps, la distance entre les nœuds est estimée en fonction de leur nombre de sauts. Les informations de distance permettent de calculer les chemins les plus courts entre toutes les paires de nœuds. Les valeurs des chemins les plus courts sont utilisées pour générer une matrice de distance D comme suit:

$$D = \begin{bmatrix} 0 & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & 0 & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & 0 & \cdots & d_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{n1} & d_{n2} & d_{n3} & \cdots & 0 \end{bmatrix} \quad (II, 12)$$

Où les lignes et les colonnes représentent les différents nœuds distribués, d_{ij} est la distance euclidienne entre le nœud i et le nœud j .

Etape 2: Dans la deuxième étape, une matrice relative est générée selon les étapes suivantes:

- Calculer la matrice J : $J = I - e \times e^T / n$ Où $e = (1, 1, 1, \dots, 1)$;
- Transformer la matrice de distance D en sa matrice de Gram H en supposant que le centre géométrique des coordonnées du nœud sont à l'origine. H est donné par:

$$H = -\frac{1}{2}JD^2J$$

- Calculer la matrice de décomposition : $H = UVU^T$

Nous désignons la matrice des plus grandes valeurs propres par V_i et U_i les i premières colonnes de U .

La matrice de coordonnées est : $X = U_i V_i^{\frac{1}{2}}$

Etape 3: Dans la phase finale, les coordonnées absolues de tous les nœuds du réseau sont déterminées.

II.6.5 SumDistMinMax

Dans cette méthode [Savvides et al, 2002], chaque nœud ancre commence par diffuser un message contenant sa position. Lorsqu'un nœud reçoit ce message de position, il estime la distance qui le sépare de cette ancre en appliquant la technique SumDist c'est une simple opération d'estimation de distance entre deux nœuds. Elle consiste à ajouter les distances mesurées entre chaque paire de nœuds voisins séparant l'ancre et le nœud à localiser. Lorsqu'une ancre envoie sa position, elle initialise le compteur de SumDist par 0. A la réception de ce message, chaque capteur voisin mesure la distance avec l'émetteur, l'enregistre, ajoute cette distance à celle contenue dans le message et fait suivre la position de l'ancre avec la distance mise à jour.

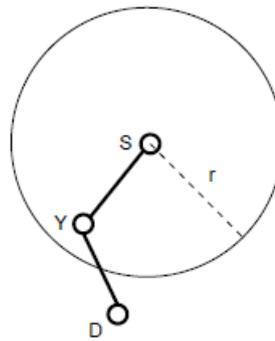


Figure II.10: Principe de SumDist [thèse].

Enfin, chaque nœud obtient les coordonnées de chacune des ancrs et calcule une estimation de la distance qui le sépare de celle-ci. La **Figure II.10** illustre le fonctionnement de la technique SumDist. Sur cette figure, la distance calculée entre les nœuds S et D est égale à $d_{SY} + d_{YD}$.

Soit x_1, x_2, \dots, x_q , est le chemin entre le nœud x_1 et l'ancre a , l'estimation de la distance entre x_1 et a (notée $\hat{d}_{x_1 a}$) est définie de façon récursive comme suit :

$$\hat{d}_{x_1 a} = d_{x_1 x_2} + \hat{d}_{x_2 a} \quad (II, 13)$$

Après cette phase d'estimation des distances avec les ancrs, les capteurs calculent leurs positions estimées en utilisant la méthode MinMax. Le principe de cette technique est de déterminer, pour chaque nœud, une « boîte » le contenant dont le centre de gravité correspond à sa position estimée.

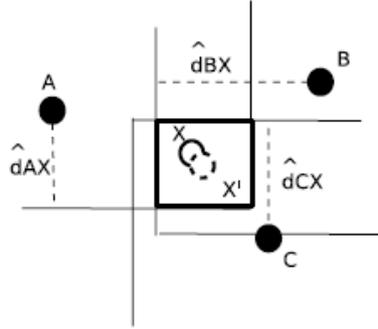


Figure II.11: Principe de MinMax [Saad, 2008].

Sur la **Figure II.11**, le nœud X associe une « boîte » à chacune des ancrs A, B, C. Chaque « boîte » est centrée en la position de l'ancr ((x_A, y_A)) et dépend de la distance estimée (\hat{d}_{XA}) avec le nœud X. Par exemple, la « boîte » centrée en A est construite comme suit :

$$[x_A - \hat{d}_{XA}, y_A - \hat{d}_{XA}] \times [x_A + \hat{d}_{XA}, y_A + \hat{d}_{XA}] \quad 2.2$$

L'intersection de ces « boîtes » forme une nouvelle « boîte » définie par :

$$[\max(x_i - \hat{d}_{Xi}), \max(y_i - \hat{d}_{Xi})] \times [\min(x_i + \hat{d}_{Xi}), \min(y_i + \hat{d}_{Xi})], i \in \{A, B, C\} \quad 2.3$$

Le capteur X estime sa position (notée X') comme étant le centre de gravité de cette «boîte».

Enfin, chaque capteur exécute un processus de raffinement où les relations de voisinage et les mesures de distances entre voisins sont prises en compte afin de faire tendre sa position estimée vers sa position réelle. L'inconvénient de cette méthode est l'accumulation des erreurs de mesure notamment dues à la technique d'estimation des distances SumDist et au processus de raffinement.

II.6.6 DV-Euclidien

La méthode DV-Euclidien [33] repose sur la propagation de la distance euclidienne séparant le nœud de l'ancr. Soit A un nœud inconnu du réseau ; pour déterminer sa position, il doit recevoir les coordonnées d'au moins deux de ses voisins : B et C. (**Figure II.12**). « A » doit également estimer les distances AB et AC par RSSI, et BC qui peut être déduite à partir de la reconstitution du voisinage de A. Si on considère le quadrilatère ABCL, où les cotés sont tous connus, la diagonale BC l'est aussi, on peut donc calculer AL

qui représente la distance euclidienne de A à l'ancre L, par application du théorème de Pythagore.

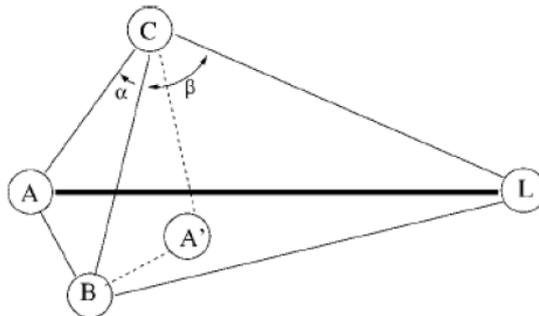


Figure II. 12 Principe de l'algorithme DV-Euclidien.

Lors du déploiement des nœuds, certains cas ambigus peuvent exister. Le conflit du choix entre la position A et A' en est un exemple dans la **Figure II.12** où les distances AB et AC sont égales respectivement aux distances $A'B$ et $A'C$. Afin de résoudre ce problème en choisissant l'une ou l'autre des positions (A ou A'), la technique DV-euclidien envisage un mécanisme de vote ou d'analyse des voisins communs. Ainsi, le nœud A décide de la distance qui le sépare de L par vote dans le cas où il dispose de plus d'une paire de nœuds voisins, ou par analyse des voisins communs aux nœuds B et C.

Cette technique présente l'avantage de permettre aux nœuds d'estimer leurs distances vers les ancres avec une meilleure précision. Par contre, son inconvénient réside dans le fait qu'elle nécessite un fort degré de connectivité pour assurer la réalisation des quadrilatères.

II.6.7 DV- Hop

DV-hop est un algorithme de localisation proposé par Dragos Niculescu et Badri Nath [Niculescu et Nath, 2003]. En DV-Hop, trois étapes sont envisagées pour rechercher les positions des nœuds inconnus, décrites comme suit:

Étape 1: au départ, toutes les ancres diffusent leurs emplacements aux autres nœuds du réseau. Les messages sont propagés saut par saut où un compteur de sauts est inclus dans le message. Chaque nœud maintient une table d'informations sur les ancres et compte le nombre minimum de sauts qui le sépare de chaque ancre.

Étape 2: lorsqu'un nœud ancre reçoit un message d'un autre nœud ancre, il estime la distance moyenne en termes de sauts en utilisant les emplacements de deux ancres et le

compteur de sauts, et le renvoie au réseau comme facteur de correction. Une ancre avec les coordonnées (x_i, y_i) , estime la distance moyenne (taille de saut i) en utilisant l'équation suivante :

$$HopSize_i = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{i \neq j} h_{ij}} \quad (II, 14)$$

où (x_j, y_j) sont les coordonnées de l'ancre j , h_{ij} est le nombre de sauts séparant l'ancre i a l'ancre j . Ensuite, après que tous les nœuds inconnus ont reçu la taille de saut des ancres, ils calculent la distance (d_i) aux différentes ancres en fonction de la taille du saut et de h_{id} (le nombre de sauts minimum), en utilisant l'équation suivante :

$$d_i = h_{id} \times HopSize_i \quad (II, 15)$$

Étape 3: La méthode de multilatération est utilisée, afin de calculer la position du nœud inconnu en fonction de la distance à chaque nœud ancre obtenue à l'étape 2. Soit (x_i, y_i) les coordonnées de l'ancre i , l'emplacement du nœud inconnu $X(x, y)^T$ peut être trouvé en utilisant la formule suivante :

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \\ (x_2 - x)^2 + (y_2 - y)^2 = d_2^2 \\ \dots \\ (x_n - x)^2 + (y_n - y)^2 = d_n^2 \end{cases} \quad (II, 16)$$

Eq. (II, 16) Peut s'écrire dans l'équation linéaire suivante: $AX = b$ (II, 17)

Où:

$$A = -2 \times \begin{bmatrix} x_1 - x_n & y_1 - y_n \\ x_2 - x_n & y_2 - y_n \\ \dots & \dots \\ x_{n-1} - x_n & y_{n-1} - y_n \end{bmatrix} \quad (II, 18)$$

$$b = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ x_2^2 - x_n^2 + y_2^2 - y_n^2 + d_n^2 - d_2^2 \\ \dots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix} \quad (II, 19)$$

$$X = \begin{bmatrix} x \\ y \end{bmatrix} \quad (II, 20)$$

L'estimation en moindres carrés de X est:

$$X = (A^T A)^{-1} A^T B \quad (II, 21)$$

Une partie de notre contribution s'intéresse particulièrement et de près à l'algorithme DV-Hop. Pour ce la et dans les paragraphes suivant nous illustrons le principe de DV-Hop par un exemple de calcul.

Exemple d'application du DV-Hop

Cet exemple illustre les calculs de DV-Hop afin de localiser un nœud inconnu. La **Figure II.13** montre le schéma d'un réseau de neuf capteurs dont trois sont des ancres. Les nœuds A, B et C sont les ancres, et S le nœud voulant calculer sa position.

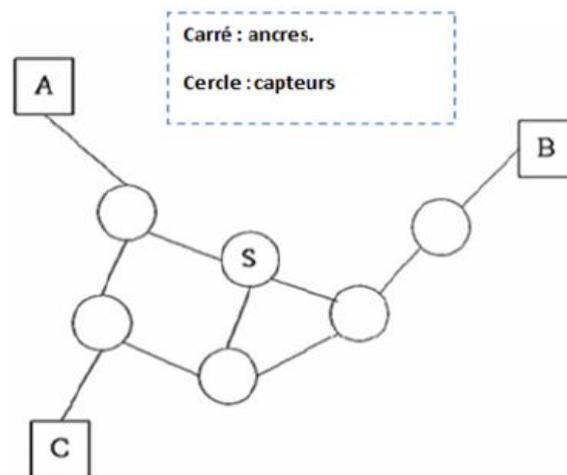


Figure II.13– Exemple d'algorithme DV-Hop.

Selon le schéma du réseau, nous avons l'information sur les distances ainsi que le nombre de sauts entre les différentes ancres dans le tableau suivant :

Tableau II-3 Valeurs illustrative pour l'exemple de DV-HOP.

	Nombres minimale de sauts séparant les ancres	Distance entre les ancres
A-B	5	120 m.
A-C	3	80 m.
B-C	5	160 m.

Chaque ancre calcule la correction (Hopsize) et la diffuse.

Ancre	Hopsiz
A	$\frac{120 + 80}{5 + 3} = 25$
B	$\frac{120 + 160}{5 + 5} = 28$
C	$\frac{160 + 80}{5 + 3} = 30$

Un nœud quelconque du réseau obtient une mise à jour de l'ancre le plus proche. La diffusion des corrections à travers le réseau est contrôlée. Dans notre exemple, le nœud S peut estimer la distance le séparant à A, B, et C à travers la mise à jour de correction reçue, A est l'ancre le plus proche du S. La distance moyenne d'un saut de A est utilisée pour l'estimation de la distance qui le sépare des ancres A, B et C.

Ancre	Distance
A	$25 \times 2 = 50$
B	$25 \times 3 = 75$
C	$25 \times 3 = 75$

A partir de ces trois valeurs, le nœud S peut déterminer sa position relative en appliquant la trilatération (voire le principe dans les paragraphes précédent), S obtiendra sa position en résolvant le système suivant :

$$\begin{cases} 50^2 = (x_s - x_A)^2 + (y_s - y_A)^2 \\ 75^2 = (x_s - x_B)^2 + (y_s - y_B)^2 \\ 75^2 = (x_s - x_C)^2 + (y_s - y_C)^2 \end{cases} \quad (II, 21)$$

II.6.8 DV-Distance

Cette méthode est similaire à la précédente (DV-Hop). Sauf que, DV-Distance remplace les sauts en DV-Hop par la distance cumulé calculée par le modèle RSSI.

L'algorithme de localisation DV-Distance est proposé par Niculescu dans [Niculescu et Nath, 2001]. Il utilise la méthode similaire au routage vectoriel de distance pour obtenir la distance cumulée. Chaque nœud inconnu obtient une distance cumulée d'au moins trois ancres, puis il utilise la trilatération pour calculer sa position. La **Figure II.14** montre le principe de DV-Distance, les nœuds 1, 2 et 3 sont des ancres et le nœud 0 est un nœud inconnu qui a besoin d'obtenir sa position. L'algorithme DV-Distance peut être divisé en trois étapes :

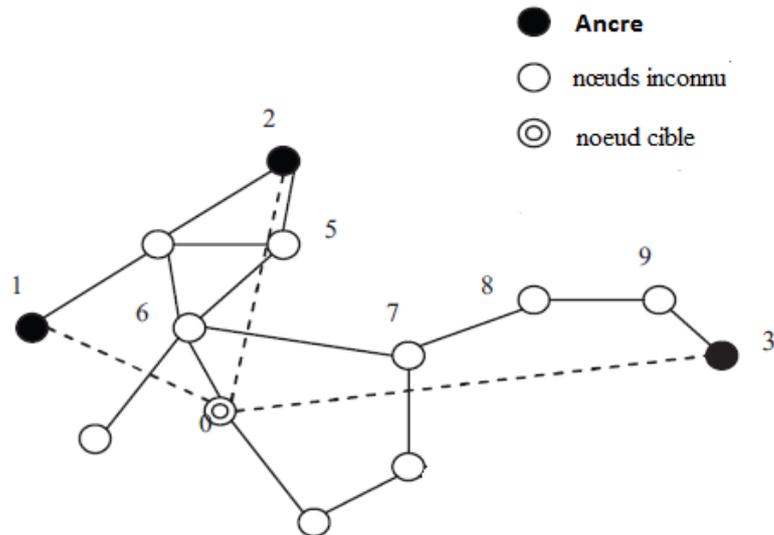


Figure II.14 Principe de la méthode DV-Distance.

Dans la première étape, chaque ancre diffuse un message "Beacon" peut être inondé à travers le réseau contenant sa position, son ID, et le RSSI correspondant. Chaque nœud récepteur calcule la distance entre ses nœuds adjacents en utilisant le modèle RSSI. Ensuite, ils compteront la distance cumulée (dis_{ij}) de lui-même aux ancres qu'ils peuvent recevoir en cumulant la distance entre les sauts.

Dans la deuxième étape, une fois qu'un nœud ancre i obtient la distance cumulée à un autre nœud ancre j , il calcule un rapport de correction (cor_{ij}) de la distance cumulé (dis_{ij}) entre les deux ancres i, j en fonction de l'équation (II.21), qui est ensuite inondée à l'ensemble du réseau en tant que distance de correction. Lorsqu'un nœud inconnu k accepte l'information d'un nœud ancre j , il utilise la distance cumulée (dis_{kj}) des nœuds ancres pour calculer la distance modifiée ($DisCorr_{kj}$) entre lui-même et l'ancre par l'équation (II.22) suivante:

$$cor_{ij} = \frac{dis_{ij}}{Dis_{ij}} \quad (II, 22)$$

$$DisCorr_{kj} = \frac{dis_{kj}}{correction} \quad (II, 23)$$

Dans la dernière étape, lorsque les nœuds inconnus obtiennent trois ou plus de distances à des ancres, les nœuds inconnus calculent leurs emplacements par trilatération ou multilatération selon le cas.

II.6.9 Bilan sur les caractéristiques des algorithmes de localisation

Le Tableau II-3 résume les principales caractéristiques des méthodes de localisation présentées dans les paragraphes précédents.

Tableau II-4 Caractéristiques des algorithmes de localisation dans les RCSFs.

caractéristique Méthode	Anchor-free	Anchor-based	Range-free	Range-based	Centralisé	Distribué
Bounding Box		#		#	*	#
APIT		#	#		*	#
GPS-LESS		#	#		*	#
MDS-MAP	#			#	#	
SumDistMinMax		#	#		*	#
DV-Distance		#		#	*	#
DV-HOP		#	#		*	#
DV-Euclidean		#		#	*	#

* signifie qu'on peut modifier la méthode pour qu'elle l'ait cet caractéristique.

II.7 Conclusion

Le problème de localisation a été identifié comme une des problématiques majeures en développement et utilisation des réseaux de capteurs sans fil. Le champ des applications qui ont besoin des informations sur la position ne cessent de s'élargir.

Dans ce chapitre, le problème de localisation dans les RCSFs a été introduit d'une manière didactique. Il commence par une introduction qui montre l'importance de la localisation, et par la suite le système de localisation, ses contraintes, ses propriétés et les trois principales phases (estimation des distances, dérivation de la position et l'algorithme de localisation) sont présentées. Il se termine par donner un aperçu des méthodes de localisation déjà existantes.

Comme nous l'avons souligné auparavant, la précision obtenue par les différentes méthodes de localisation traditionnelles n'est pas suffisante pour toutes les applications, d'où la nécessité de l'améliorer. Pour ce la, Certains chercheurs développent de nouveaux modèles, pour optimiser les systèmes de localisation afin de minimiser l'erreur sur l'estimation de la position. Ces modèles s'inspirent en large partie de l'observation des systèmes naturels et en particulier du comportement de groupes et d'animaux sociaux. Parmi les systèmes artificiels collectifs à fonctionnalité émergente, on trouve les algorithmes évolutionnaires et leurs variantes, les réseaux de neurones formels ainsi que

les algorithmes inspirés des comportements collectifs des animaux sociaux tels que les algorithmes d'optimisation par essaim et les algorithmes de fourmis artificielles.

Nos travaux de thèse sont basés sur l'utilisation des méthodes bio-inspirées pour optimiser la localisation dans les RCSFs. Pour cela, le prochain chapitre est consacré à la présentation de cette branche des méthodes d'une façon générale et en détail, l'algorithme d'optimisation par les fruits Flay (FOA) utilisé dans nos contributions.

-

Chapitre III

Des métaheuristiques d'optimisation

Chapitre III

Des métaheuristiques d'optimisation

III.1. Introduction

Dans notre vie quotidienne, nous sommes fréquemment confrontés à des problèmes plus au moins complexes. Cela peut commencer au moment où l'on tente de ranger notre bureau, de placer nos mobiliers, de gérer l'espace dans notre maison, de minimiser nos itinéraires et aller jusqu'à un processus industriel, on définit alors une fonction objective, que l'on cherche à optimiser par rapport aux différentes contraintes.

L'optimisation est un sujet central en ingénierie. Plusieurs problèmes industriels peuvent être décrits sous la forme de problèmes d'optimisation, qui ont pour objectif principal de trouver une solution qui permet d'optimiser (minimiser ou maximiser) une fonction objective appelée dans la littérature *fitness* tout en vérifiant un certain nombre de contraintes.

En informatique, lorsqu'un problème se pose il faut concevoir de nouveaux algorithmes de résolution car les techniques existantes ne nous permettent pas d'aboutir aux résultats souhaités. Ainsi, lorsque l'on veut définir une nouvelle méthode de résolution d'un problème, il faut souvent une source d'inspiration afin de garantir des solutions satisfaisantes. Les recherches sur les comportements collectifs des insectes sociaux fournissent aux informaticiens des méthodes puissantes pour la conception d'algorithmes d'optimisation difficile appelée **méta-heuristiques**.

Les méta-heuristiques forment une famille d'algorithmes visant à résoudre des problèmes d'optimisation difficile, pour lesquels on ne connaît pas de méthode classique plus efficace. Elles sont aussi utilisées comme des méthodes d'amélioration pouvant optimiser une large gamme de problèmes différents, sans nécessiter de changements profonds dans l'algorithme employé.

Le présent chapitre vise une vue d'ensemble sur l'optimisation globale et les méta-heuristiques de résolution.

III.2. Quelques notions générales sur l'optimisation

D'une façon générale, un problème d'optimisation est défini par un ensemble de solutions probables S_i , dont leur qualité peut être vérifiée par une fonction objective f_{obj} . Alors, le processus d'optimisation cherche à trouver la meilleure solution S^* possédant une qualité optimal trouver par $f_{obj}(S^*)$ en vérifiant certaines contraintes d'égalité ou d'inégalité sur S [Allaoua et Loutfi, 2009].

Par la suite, on peut chercher à minimiser ou à maximiser $F(S)$. Sans perte de généralité, nous nous plaçons désormais dans le cadre d'un problème de la recherche des minimums d'une fonction, puisqu'un problème de maximisation peut toujours se ramener à un problème de minimisation.

Un problème de minimisation de dimension n peut être formulé dans le cas général comme suite :

$$\left\{ \begin{array}{l} \min f_{obj}(x) \\ x \in D \\ g_i(x) \leq 0 \quad 1 \leq i \leq n \\ h_i(x) = 0 \quad 1 \leq j \leq n \end{array} \right. \quad (\text{III.1})$$

Où :

- f_{obj} : la fonction à minimiser.
- x : Est un vecteur à n dimension représentant les objets (variables) du problème traité.
- D : Est l'espace de recherche.
- $g_i(x) \leq 0, h_i(x) = 0$: Les contraintes d'inégalité et d'égalité. Ces contraintes permettent de restreindre le domaine de validité de la solution à D . Notons que ces contraintes peuvent ne pas exister, auquel cas nous parlons d'optimisation sans contraintes.

La figure (**Fig.3.1**) présente un exemple de distribution possible des optimums (minimum et maximum) d'une fonction objectif.

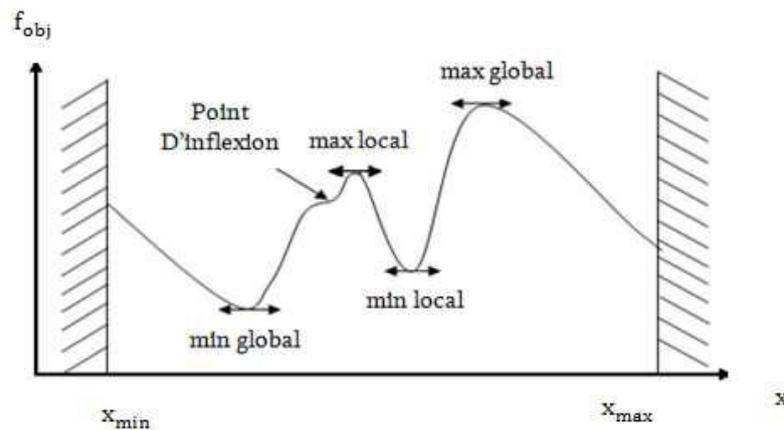


Figure III.1 Point singuliers d'une fonction objective [Benyamina, 2013].

Dans la littérature, les techniques proposées pour résoudre les problèmes d'optimisation globale sont réparties en différentes classes. Selon les auteurs de [Törn et Zilinskas, 1989], les techniques d'optimisation se divisent en deux grandes parties : l'optimisation discrète (combinatoire) et l'optimisation continue, la figure suivante (**Fig.3.2**) montre une classification générale des méthodes d'optimisation globale.

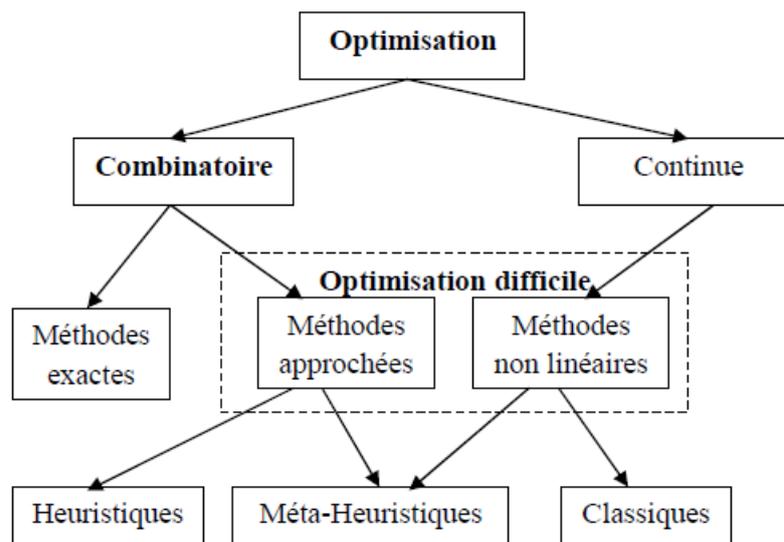


Figure III.2 Classification des méthodes d'optimisation [benyamina, 2013].

Pour la résolution des problèmes d'optimisation, on a recours aux méthodes approchées, lorsqu'on est confronté à un problème difficile; dans ce cas, le choix est parfois possible entre une heuristique, entièrement dédiée au problème traitée, et une méta-heuristique [Berthiau et Siarry, 2001].

L'adaptation des méta-heuristiques à l'optimisation est un sujet de recherche en soi; en effet, cette adaptation pose un problème spécifique pour chaque méta-heuristique, et les démarches publiées sont variées. Ces méthodes utilisent cependant un haut niveau d'abstraction, leur permettant d'être adaptées à une large gamme de problèmes différents [Berthiau et Siarry, 2001].

III.3. Littérature des méta-heuristiques d'optimisation existantes

Au cours des trente dernières années, un grand intérêt a été manifesté pour l'utilisation des méta-heuristiques pour la résolution des problèmes d'optimisation combinatoire. Dans ce paragraphe, nous pouvons essayer d'indiquer les principales étapes qui ont marqué l'histoire des méta-heuristiques.

La contribution pionnière a été la proposition de la méthode de recuit simulé (abrégé - RS) par Kirkpatrick et al. en 1983 [Kirkpatrick et al, 1983]. En 1986, l'algorithme de recherche tabou (RT) a été proposé par Glover [Glover, 1986], et l'algorithme AIS (système immunitaire artificiel) est proposé par Farmer et al. [Farmer et al, 1992]. En 1988, Koza a déposé son premier brevet sur la programmation génétique, publié plus tard en 1992 [Koza, 1992]. En 1989, Goldberg a publié un livre bien connu sur les algorithmes génétiques (AG) [Goldberg, 1989]. En 1992, Dorigo a terminé sa thèse de doctorat, dans laquelle il décrit ses travaux innovants sur l'optimisation par des colonies de fourmis (ACO) [Dorigo, 1992]. En 1993, le premier algorithme basé sur les colonies d'abeilles a été proposé par Walker et al. [Walker et al, 1993]. Une autre étape importante c'est le développement de la méthode d'optimisation des essaims de particules (PSO) par Kennedy et Eberhart en 1995 [Kennedy et Eberhart, 1995]. Dans la même année, Hansen et al. ont proposé CMA-ES (Covariance Matrix Adaptation Evolution Strategy) [Hansen et al, 1995] et [Feo et Resende, 1989] ont proposé la méthode GRASP (Greedy Randomized Adaptive Search Procedure). En 1996, l'algorithme à estimation de distribution est proposé par Mühlenbein et Paaß [Mühlenbein et Paaß, 1996]. En 1997, Storn et Price ont proposé l'algorithme d'évolution différentielle [Storn et Price, 1997]. En 2001, Geem et al, s'inspirant du processus d'improvisation des interprètes de musique, ont proposé l'algorithme de recherche d'harmonie [Geem et al, 2001]. En 2002, Passino a introduit un algorithme d'optimisation basé sur le comportement sociale des bactéries *Escherichia coli* pour la recherche de nourriture [Passino, 2002]. En 2006, He et al. ont proposé un nouvel algorithme d'optimisation basé sur la population, appelé l'optimiseur de recherche de

groupe, qui était basé sur le modèle producteur-scrutateur [He et al, 2006]. En 2008, Simon a proposé un algorithme d'optimisation basé sur la biogéographie [Simon, 2008], fortement influencé par la théorie de l'équilibre de la biogéographie insulaire [MacArthur et Wilson, 1967]. Dans cette année, le principe d'attraction entre les lucioles est inspiré par Dr Xin-She Yang à l'université de Cambridge pour développer l'algorithme de Firefly [Yang, 2008]. En 2009, Xin-She Yang et Suash Deb ont proposé une recherche coucou [Yang et Deb, 2009]. Cet algorithme est basé sur le comportement de reproduction de certaines espèces de coucous en combinaison avec le comportement de vol de Lévy de certains oiseaux et mouches des fruits. Dans la même année, Rashedi et al. [Rashedi et al, 2009] ont proposé l'algorithme de recherche gravitationnelle basé sur une simulation du comportement de la force gravitationnelle de Newton. En 2010, Yang [Yang, 2010] a proposé une méthode méta-heuristique inspiré des chauves-souris, basé sur le comportement d'écholocation des chauves-souris. En 2011, le professeur Taïwan "Wen-Tsao Pan" dans [Pan, 2011], simule le comportement de recherche de nourriture par la drosophile (Fruit Fly) pour inventé un modèle efficace, presque sans paramètre spécial et peut atteindre rapidement une optimisation précise appelé FOA (algorithme d'optimisation par Fruit Fly).

Nous présentons dans ce chapitre une collection non exhaustive de quelques méta-heuristiques. L'omission de certains algorithmes ne signifie pas qu'ils ne sont pas populaires, mais il n'est pas possible d'inclure tous les algorithmes. Dans ce qui suit, nous présentons:

- Le recuit simulé (SA) ;
- Les algorithmes génétiques (AG);
- L'optimisation par des colonies de fourmis (ACO) ;
- L'optimisation par des essaims de particules (PSO);
- L'évolution différentielle (DE);
- L'algorithme de recherche coucou (CS);
- L'algorithme des colonies d'abeilles (ABC);
- L'algorithme de FireFly (FFA);
- L'optimisation par essaim de poulets (CSO) ;
- L'algorithme de Fruit Fly (FOA).

III.3.1 Le recuit simulé

La méta-heuristique de recuit simulé (SA- Simulated Annealing en anglais) est une méthode empirique inspirée d'un processus utilisé en métallurgie. Les origines du recuit simulé remontent aux expériences réalisées par Métropolis et al. dans les années 50 pour simuler l'évolution d'un processus de recuit physique, afin de comprendre le fonctionnement d'un système thermodynamique. L'idée est de faire une analogie avec un processus industriel de fabrication d'alliages métalliques qui consiste à refroidir le métal en fusion par paliers de température successifs de façon à lui assurer une structure sans défaut [Metropolis et al, 1953].

L'utilisation d'un tel processus du recuit simulé pour résoudre des problèmes d'optimisation combinatoire a été mise au point par trois chercheurs de la société IBM, S. Kirkpatrick, C.D. Gelatt et M.P. Vecchi en 1983 dans [Kirkpatrick et al, 1983], et indépendamment par V. Cerny en 1985 [Černý, 1985]. Le principe de l'algorithme SA, fut obtenu par analogie avec le phénomène thermodynamique de recuit des métaux. Initialement, le métal est porté à très haute température, puis il est refroidi progressivement. Le recuit simulé s'appuie sur l'algorithme de **Metropolis-Hastings**, qui permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie **E** du système. On introduit également un paramètre fictif, la température **T** du système. Sur la base de ce principe, les principales étapes de recuit simulé sont données ci-après (Fig. III.3):

1. Définir une fonction objective $f(x)$.
2. Choisir une température initial T_0 .
3. Générer une position aléatoire P_1 , considéré come une solution initial.
4. Copier la position P_1 dans une position P_2 , puis échanger deux colonnes (choisies aléatoirement) de P_1 .
5. Calculer $\Delta = f(P_2) - f(P_1)$
6. Si $\Delta \leq 0$, $P_1 \leftarrow P_2$; le cas échéant, mettre à jour le meilleur score (valeur de fitness) et la meilleure position. Aller à 9.
7. Générer un nombre réel aléatoire r dans l'intervalle $[0,1]$.
8. Si $r < \exp(-\Delta / T)$, $P_1 \leftarrow P_2$.
9. Diminuer T .
10. Retourner à 4, tant qu'on n'a pas satisfait un critère de terminaison.

Figure III.3 Algorithme de recuit simulé.

III.3.2 Les algorithmes génétiques

Les algorithmes génétiques (AG) sont des algorithmes basés sur deux mécanismes biologiques qui sont les lois de Mendel et le principe fondamental de la théorie de l'évolution de Charles Darwin. Holland exposa les principes de ces algorithmes pour permettre aux ordinateurs d'imiter les principes d'évolution des êtres vivants afin de rechercher des solutions aux problèmes d'optimisation combinatoire [Holland, 1975]. Il expliqua d'abord comment ajouter de l'intelligence dans un programme informatique avec les croisements et la mutation. Cette méta-heuristique a été enrichie par les travaux de Goldberg dans [Goldberg, 1989], en s'appuyant sur le parallèle des idées suivant:

- Un individu est lié à un environnement par son code d'ADN ;
- Une solution est liée à un problème par son indice de qualité ;
- Une « bonne » solution à un problème donné peut être vue comme un individu susceptible de survivre dans un environnement donné.

Il formalisa ensuite les principes fondamentaux des algorithmes génétiques [Fonseca et Fleming, 1993] [Dreo et Siarry, 2003] :

- La capacité de représentations élémentaires, comme les chaînes de bits, à coder des structures complexes.
- Le pouvoir de transformations élémentaires à améliorer de telles structures.

Si on représente par 'P(t)' la population de solutions probable (candidates) pour un problème donné, à l'instant (itération) t. Les principales étapes d'un algorithme génétique peuvent être présentées par les quatre points illustrés sur la figure suivante :

1. Initialiser les P(t) initiaux.
2. Évaluer chaque individu de P(t).
3. **Tant que**, un critère d'arrêt ou un maximum d'itération n'est pas satisfait.
 - $t = t + 1$.
 - Sélectionner P(t+1) à partir P(t).
 - Croisement de P(t+1).
 - Muter P(t+1).
 - Évaluer P(t+1).
4. **Fin tant que**
4. Retourner la meilleure solution parmi les p(t) de dernière itération.

Figure III.4 Principe de l'Algorithme génétique.

III.3.3 Optimisation par des colonies de fourmis

L'algorithme ACO (Ant Colony Optimization en Anglais), est une méta-heuristique d'optimisation inventé par Dorigo dans [Dorigo et Stutzle, 2004] [Dorigo, 2007], et qui est inspiré du comportement de recherche de la nourriture des colonies de fourmis dans la nature.

L'idée originale repose sur l'observation de l'exploitation des ressources alimentaires chez la colonie de fourmis naturelles. En effet, les fourmis ont des capacités cognitives individuellement limitées, et elles sont capables collectivement de trouver le chemin le plus court entre leur nid et une source de nourriture. En effet, chaque fourmi laisse derrière elle une traînée chimique appelé « phéromone » pour attirer l'attention de ses collègues. Les fourmis choisissent les chemins contenant les plus fortes concentrations de phéromone. Il se forme ainsi des autoroutes qui sillonnent le paysage [Laptik et Navakauskas, 2007]. Ce mode de communication particulier est appelé stigmergie.

Le premier modèle artificielle utilisant ce comportement biologique est conçu pour résoudre le problème du voyageur de commerce. Le principe consiste à lancer des fourmis, et à les laisser construire pas à pas le meilleur itinéraire pour aller d'une ville à l'autre. Au début, plusieurs chemins de longueurs différentes sont possibles, et les fourmis les empruntent tous en se dirigeant aléatoirement. Les traces de phéromone se concentrent assez rapidement sur l'itinéraire optimal. C'est donc un algorithme qui repose sur la construction progressive de solutions [Dorigo et Caro, 1999] [Benyamina, 2013].

Le principe de recherche par ACO est présenté sur la Figure II.5.

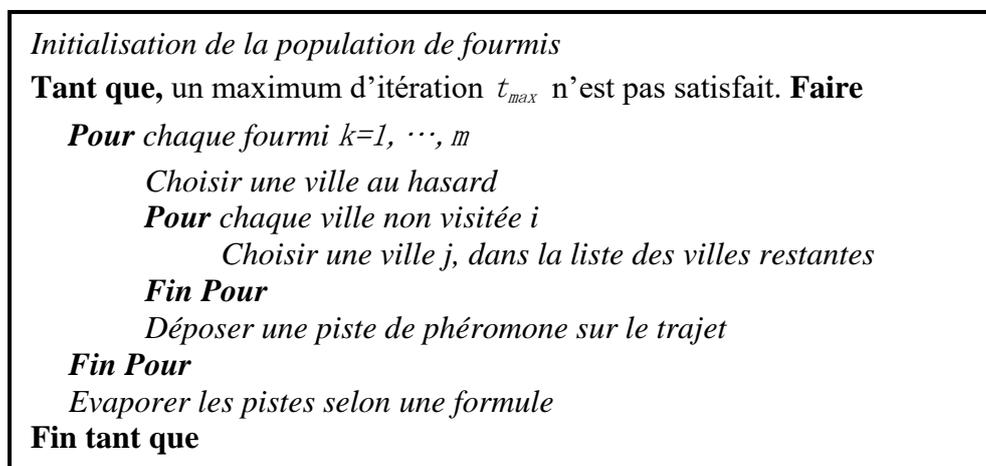


Figure III.5 Algorithme de colonies de fourmis.

III.3.4 L'optimisation par des essais de particules

L'optimisation par des essais de particules est une technique informatique évolutive développée par Eberhart et Kennedy en 1995 [Kennedy et Eberhart, 1995]. Elle appartient à la classe des algorithmes d'optimisation stochastique qui simule le comportement social du vol d'oiseaux. L'algorithme PSO est facile à implémenter et il est efficace en termes de calcul car ses besoins en mémoire et en CPU sont faibles.

La technique PSO utilise un ensemble de solutions réalisables appelées «particules» qui sont déployé dans l'espace de recherche avec des emplacements initiaux aléatoires. Les valeurs de la fonction objective correspondant aux emplacements des particules sont évaluées. Ensuite, les particules sont déplacées dans l'espace de recherche en obéissant à des règles inspirées du comportement de vol des oiseaux [Kennedy et Eberhart, 1995]. Chaque particule est déplacée vers la moyenne pondérée de façon aléatoire de la meilleure position que la particule a rencontrée jusqu'à présent (P_{best}) et de la meilleure position rencontrée par l'ensemble de la population de particules (g_{best}).

Soit $X_i = (x_{i1}, x_{i2}, \dots, x_{iM})$ le vecteur de N dimension représentant la position de la $i^{\text{ème}}$ particule dans l'essaim, $g_{best_i} = [g_1, g_2, \dots, g_n]$ est le vecteur de position de la meilleure particule dans l'essaim (c'est-à-dire la particule avec la plus petite valeur de fonction objective). $P_{best_i} = [p_{i1}, p_{i2}, \dots, p_{iN}]$ est le vecteur de position de la $i^{\text{ème}}$ particule et la vitesse de la particule $V_i = [v_{i1}, v_{i2}, \dots, v_{iN}]$ soit la vitesse de la $i^{\text{ème}}$ particule. Les particules évoluent selon les équations:

$$P_{i,j}(t+1) = P_{i,j}(t) + V_{i,j}(t+1) \quad (\text{III.2})$$

$$v_{i,j} = k \left[w * v_{i,j}(t) + c_1 * r_1 * (I_{i,j} - v_{i,j}(t)) + c_2 * r_2 * (g_j - v_{i,j}(t)) \right] \quad (\text{III.3})$$

Avec :

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \quad \varphi = c_1 + c_2 \text{ et } \varphi > 4, \text{ avec } j = 1, \dots, N,$$

w : est une constante appelée facteur d'inertie.

c_1 et c_2 : sont des constantes appelées coefficients d'accélération.

r_1 et r_2 sont des nombres aléatoires uniformément distribués dans l'intervalle $[0,1]$.

Si la vitesse calculée fait sortir une particule de l'espace de recherche, sa fitness n'est pas calculée. Compte tenu de la nouvelle vitesse, obtenue à partir de p_{best_i} et g_{best_i} .

III.3.5 Algorithme de l'Evolution Différentielle

La méta-heuristique d'évolution différentielle (Differential Evolution en anglais) est une méthode bio-inspiré proposée par Storn & Price en 1997 [Storn et Price, 1997], et appartient à la classe des algorithmes évolutionnaires (Figure III.6), lesquels s'inspirent de la théorie de l'évolution des espèces dans la nature (Figure III.7).

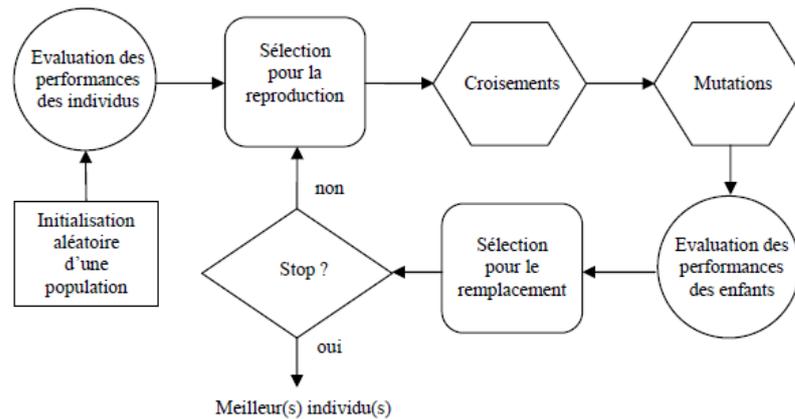


Figure III.6 Principe des algorithmes évolutionnaires [Feyel, 2015]

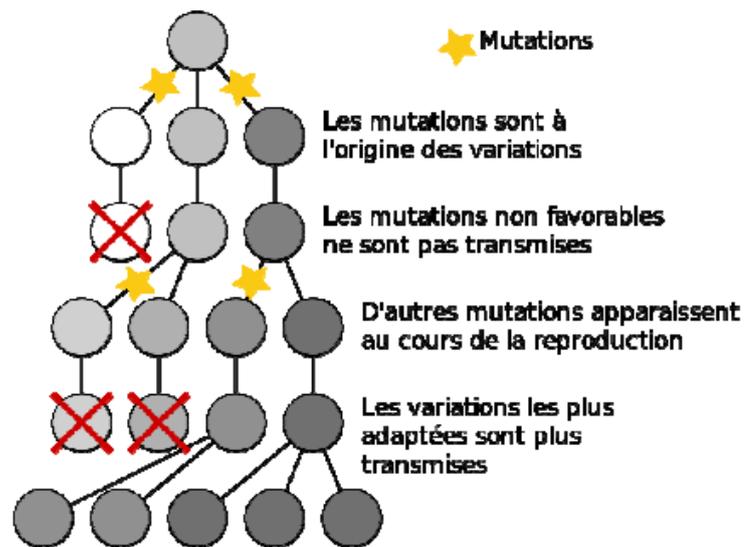


Figure III.7 Théorie de Darwin [Feyel, 2015]

La DE est un algorithme utilisée pour la résolution des problèmes d'optimisation des fonctions multidimensionnelles. Elle est très efficace dans les cas où les paramètres ne peuvent pas être codés comme des vecteurs réels. L'idée principale de cet algorithme réside sur l'utilisation d'un opérateur de recombinaison ternaire pour la création de nouveaux individus. Donc, la différence entre chaque deux individus est ajoutée à un troisième autre

individu sur la base de population pour créer des individus de la nouvelle génération [Onwubolu et Babu, 2004] [Price et al, 2005].

Le modèle de base d'évolution différentielle a beaucoup évolué et plusieurs améliorations ont été proposées [Cuevas et al, 2010]. L'algorithme DE marque la différence avec les AGs (Algorithmes Génétiques) en donnant plus d'importance à la mutation qui est le résultat d'une petite perturbation dans les gènes du chromosome. Cette mutation a pour objectif de créer un nouvel individu (solution) à partir des différences entre deux individus de la population. Le croisement consiste à modifier chaque gène de l'individu par celui du mutant avec une certaine probabilité. Le nouveau individu créé est sélectionné pour remplacer une solution de la population si sa fitness est meilleure.

Dans le pseudo code de l'algorithme DE présenté sur la Figure III.8, les opérateurs de mutation, croisement et sélection sont répétés pour chaque solution de la population.

```

1. générer une population  $X$  de solution initiale
2. Evaluer chaque individu de la population
3. Fixer le nombre maximum des générations
   Pour tous les individus  $j$  de la population  $X$ 
     Création du vecteur de différence
     Générer aléatoirement des solutions  $l,m,n$ 
     Calculer le vecteur de solution  $Y$ 
      $y_k = X_m + F(X_n - X_l)$ 
     Croisement
     Générer un nouveau vecteur de solution  $C$ 
     Évaluer le vecteur de solution  $C$ 
     Sélection
     Si  $F(C) < F(X_j)$ 
        $X_j = C$ 
        $F(C) = F(X_j)$ 
     Fin si
   Fin de pour
4. visualisation des résultats de la meilleure solution de la population.
```

Figure III.8 Algorithme de l'évolution différentielle.

III.3.6 Algorithme de la recherche coucou

Les coucous sont des oiseaux fascinants, non seulement en raison des beaux sons qu'ils peuvent produire, mais aussi en raison de leur stratégie de reproduction agressive. Certaines espèces comme les coucous Ani et Guira pondent leurs œufs dans des nids communaux, bien qu'elles puissent en retirer d'autres pour augmenter la probabilité d'éclosion de leurs propres œufs [Yang et Deb, 2009]. Un grand nombre d'espèces s'attaquent au parasitisme obligatoire des couvées en pondant leurs œufs dans les nids d'autres oiseaux hôtes (souvent d'autres espèces). Certains oiseaux hôtes peuvent entrer en conflit direct avec les coucous envahissants. Si un oiseau hôte découvre que les œufs ne sont pas les leurs, il jettera ces œufs extraterrestres ou abandonnera simplement son nid et construira un nouveau nid ailleurs. Certains coucous ont évolué de telle manière que les coucous parasites femelles sont souvent très spécialisés dans le mimétisme des couleurs et des motifs des œufs de quelques espèces hôtes choisies. Cela réduit la probabilité d'abandon de leurs œufs et augmente ainsi leur reproductivité [Yang et Deb, 2009].

Pour simplifier la description de l'algorithme d'optimisation par recherche coucou (CSO), les trois règles idéalisées suivantes sont utilisées: (1) Chaque coucou pond un œuf à la fois et jette son œuf dans un nid choisi au hasard; (2) Les prochaines générations porteront les meilleurs nids avec des œufs de haute qualité; (3) Le nombre de nids d'hôtes disponibles est fixe, et l'œuf pondu par un coucou est découvert par l'oiseau hôte avec une probabilité (p_a). La probabilité se situe dans la plage de l'intervalle $[0, 1]$. Dans ce cas, l'oiseau hôte peut jeter l'œuf ou abandonner le nid et construire un nid complètement nouveau. Pour simplifier, cette dernière hypothèse peut être approximée par la probabilité de fraction (p_a) des nids remplacés par de nouveaux nids (avec de nouvelles solutions aléatoires). Pour un problème de maximisation, la qualité ou l'adéquation d'une solution peut simplement être proportionnelle à la valeur de la fonction objective [Yang et Deb, 2009].

Sur la base des trois règles ci-dessus, les étapes de base du CSO peuvent être résumées dans le pseudo-code suivant :

<p>Commencer fonction objective $f(x)$, $x = (x_1, \dots, x_d)^T$ Générer une population initiale de nids de taille n, x_i ($i = 1, 2, \dots, n$) Tant que ($t < \text{maximum d'itération}$) ou (critère d'arrêt) Obtenez un coucou au hasard avec les vols Lévy Évaluer sa qualité par la fitness F_i Choisissez un nid parmi N au hasard si ($F_i < F_j$), Remplacer j par la nouvelle solution; Fin si Une probabilité de fraction (P_a) des pires nids est abandonnée et un nouveau nid (solution) est construit; Conserver les meilleures solutions; Classez les solutions et trouvez le meilleur nid actuel; Fin tant que Retourner la meilleure solution. Fin</p>
--

Figure III.9 Pseudo code de l'Algorithme de recherche coucou.

Ici le concept de fitness, F_i est utilisé pour guider les vols de Lévy lors de la recherche du nid optimal (solutions) dans l'espace de recherche N -dimensionnel. D'autre part, diverses études ont montré que le comportement de vol de nombreux animaux et insectes a démontré les caractéristiques typiques des vols de Lévy [Brown et al, 2007] [Pavlyukevich, 2007]. Le vol de Lévy est défini comme une marche aléatoire avec des longueurs de pas basées sur une distribution de probabilité à queue lourde. Des études sur le comportement humain telles que les modèles de recherche de chasseurs-cueilleurs montrent également la caractéristique typique des vols Lévy. Par la suite, un tel comportement a été appliqué à l'optimisation et à la recherche optimale avec une capacité prometteuse [Shlesinger et Zaslavsky, 1995] [Shlesinger, 2006].

Lors de la génération de nouvelles solutions $x(t+1)$, par exemple pour un coucou i , un vol de Lévy est effectué comme indiqué dans l'équation suivante :

$$x(t+1) = x(t) + \alpha \oplus \text{Lévy}(\lambda) \quad (\text{III.4})$$

Où $\alpha > 0$ est la taille de pas qui doit être liée aux échelles du problème d'intérêt. Dans la plupart des cas, nous pouvons utiliser $\alpha=1$. Le vol de Lévy fournit essentiellement une marche aléatoire tandis que la longueur de pas aléatoire est tirée d'une distribution de Lévy selon l'équation suivante : $\text{Lévy} \sim u = t^{-\lambda}$ ($1 < \lambda \leq 3$) (III.5)

III.3.7 Algorithme des colonies d'abeilles

La méta-heuristique d'optimisation par des colonies d'abeilles artificielles ou « Artificial Bee Colony » en anglais (ABC) est une méthode bio-inspiré proposé par KAR dans [KARABOGA et BASTURG, 2007], qui s'inspire du comportement naturel d'une colonie d'abeilles mellifères lors de la recherche de la nourriture.

Dans la nature, les abeilles utilisent un processus coopératif pour rechercher la nourriture. Ce processus est basé sur un mécanisme très efficace pour guider les déplacements des abeilles de la colonie. Une abeille permet d'attirer l'attention d'autres abeilles de la colonie aux sources alimentaires trouvées afin de collecter des ressources diverses. En fait, ils utilisent un ensemble de danses frétilantes permettent aux abeilles de partager des informations sur la quantité, la distance et après la direction du nectar avec ses congénères. Donc, ces danses représentent le moyen de communication entre les différentes abeilles de la colonie afin de collaborer et partager la connaissance collective des abeilles sur la quantité du nectar dans la source de nourriture trouvée par chacun des membres de la même colonie [Metropolis et al, 1953].

Dans un ABC (algorithme d'optimisation par colonies d'abeilles), chaque source de nectar correspond à une solution candidat au problème à optimiser. La population d'une colonie d'abeilles artificielles est composée de trois types d'agents dont chacun à son travail:

- Les scouts : ceux lançant une recherche aléatoire d'une nouvelle source de nourriture afin d'exploiter l'espace de recherche.
- L'ouvrière mémorise exploite la source de nourriture trouvée. Et sur la base de sa mémoire, elle essaye de modifier sa solution (i.e. position) actuelle pour découvrir une nouvelle source de nourriture.
- L'abeille spectatrice : recueillir des informations sur les sources de nectar trouvé par les ouvrières.

Les principales phases après l'initialisation de l'algorithme ABC, est de faire placer les abeilles de type ouvrières puis les spectatrices sur les sources de nourriture en mémoire; et à la fin, envoyer les scouts (éclaireurs) dans la zone de recherche pour découvrir de nouvelles sources de nourriture. Cette séquence des phases est répétée jusqu'à ce que les exigences d'optimisation soient remplies.

Ces phases de la méta-heuristique d'ABC sont détaillées dans le pseudo code de la Figure III.10 suivante :

Initialisation : Pour toute les ouvrières
 Générer une solution initiale X_i ,
 Evaluer la solution par la fonction objective $f(X_i)$
 Evaluer la quantité de nectar associée,
 Initialiser un compteur d'abandon $C_i = 0$.

Tant que le critère d'arrêt n'est pas satisfait, **répéter :**

Phase ouvrière :

Pour chaque ouvrière
 Générer une nouvelle position X_{new} ,
 Evaluer par la fonction de coût correspondante $f(X_{new})$,
 Evaluer sa quantité de nectar,
Si $f(X_{new}) \leq f(X_i)$, **alors**
 $X_i = X_{new}$
 $C_i = 0$
Sinon $C_i = C_i + 1$
Fin Si

Fin Pour
 Evaluer la probabilité de transmission d'information associée à chaque source p_i ,

Phase spectatrice :

Pour chaque spectatrice **Si** $\text{rnd}(0,1) < p_i$
 Générer une nouvelle position X_{new} ,
 Evaluer la fonction de cout correspondante $f(X_{new})$,
 Evaluer sa quantité de nectar;
Si $f(X_{new}) \leq f(X_i)$, **alors**
 $X_i = X_{new}$
 $C_i = 0$
Sinon $C_i = C_i + 1$
Fin Si

Phase scoute :

Pour chaque scoute **Si** $C_i \geq T_{limit}$
 Générer une nouvelle position,
 Evaluer la fonction de cout correspondante
 Evaluer sa quantité de nectar,
 $C_i = 0$

Retenir la meilleure solution X_{best}

Fin Tant que

Figure III.10 Pseudo code de l'Algorithme ABC.

III.3.8 Algorithme de Firefly

L'algorithme de la méta-heuristique des Firefly (Lucioles) est une méthode Bio-inspirée, inventé par Dr Xin-She Yang à l'université de Cambridge en 2008 [Yang, 2008].

Cet algorithme est basé sur le principe d'attraction entre les lucioles et simule le comportement d'un essaim de lucioles dans la nature, ce qui lui donne beaucoup de similarités avec d'autres méta-heuristiques basées sur l'intelligence collective du groupe [Yang, 2009].

Le modèle artificielle de l'algorithme des Firefly prend en considération les trois règles suivantes [Yang, 2010] :

1. Toutes les lucioles sont unisexes, ce qui fait que l'attraction entre celles-ci n'est pas en fonction de leur sexe.
2. L'attraction est proportionnelle à leurs luminosités, donc pour deux lucioles, la moins lumineuse se déplacera vers la plus lumineuse. Si aucune luciole n'est lumineuse qu'une luciole particulière, cette dernière se déplacera aléatoirement.
3. La luminosité des lucioles est déterminée en fonction d'une fonction objective (à optimiser)

En se basant sur les trois points précédents, le pseudo code de l'algorithme de Firefly est montré sur la Figure III.11 [Rabhi et Fizazi, 2015].

```

Définir une fitness au problème  $f(x)$ 
Générer une population initiale des lucioles
Définir la valeur de l'intensité de lumière ( $I$ )
Définir une valeur pour le coefficient d'absorption ( $\gamma$ )
Tant que ( $t < \text{Max d'itération}$ )
  Pour  $i = 1 \dots n$ 
    Pour  $j = 1 \dots n$ 
      Si ( $I_i < I_j$ )
        La luciole  $j$  se déplacera vers la luciole  $i$ 
      Fin Si
      Varier l'attraction en fonction de la distance
      Evaluer les nouvelles positions par  $f(x)$ 
      Mettre à jour l'intensité de lumière
    Fin Pour  $j$ 
  Fin Pour  $i$ 
  Classer les lucioles et trouver la meilleure solution
Fin Tant que
Visualiser les résultats

```

Figure III.11 Pseudo code de l'Algorithme de Firefly

III.3.9 Optimisation par essaim de poulets

Les poulets sont sans limite connus comme animal domestique pour la source de nourriture. Les œufs de poulet sont également considérés comme une source de nourriture et conservés en conséquence. Le poulet est considéré comme un oiseau social qui aime vivre ensemble comme un troupeau. Ils sont intellectuellement fusionnés et ont la capacité de se reconnaître sur plus de 100 individus même s'ils sont séparés pendant plusieurs mois. Les poulets utilisent 30 sons différents et reconnaissables pour communiquer entre eux, ce qui comprend un large éventail de variétés, notamment gloussements, pépiements, nidification, découverte de nourriture, accouplement, danger, etc. Étant un animal social et un oiseau, ils ont tendance à acquérir de l'expérience et à se renseigner sur différentes situations grâce à la méthode des essais et des erreurs. En outre, ils apprennent également de leurs expériences antérieures et d'autres pour prendre leurs décisions [Meng et al, 2014].

L'ordre hiérarchique joue un rôle vital sur la vie sociale d'essaim des poulets. Le poulet dominant dans un essaim établira sa domination sur les plus faibles. Le plus dominant comptait comme le coq du groupe. Sur la base de la dominance, il y aurait des poulets plus forts qui seront considérés comme des poules et garderont une distance très proche avec le coq [Meng et al, 2014]. Les soumis sont considérés comme des poussins. Le coq se tient en marge du groupe. L'ajout ou la suppression de poulets dans un groupe entraînerait une interruption limitée de l'ordre jusqu'à ce qu'un ordre hiérarchique déterminé soit établi.

Pour accéder à la nourriture, le dominant a la priorité et c'est le devoir du coq d'appeler ses camarades de groupe pour accéder à la nourriture quand il y a une source de nourriture. Tout en élevant les poussins, le comportement social joue également un rôle vital dans un troupeau. Le coq dans un groupe émet également un appel fort lorsqu'il y a une chance d'invasion sur leur territoire [Meng et al, 2014]. Le comportement des poulets peut varier selon le sexe. Le rôle général d'un coq serait de fouiller la nourriture ainsi que de se battre avec les poulets d'autres groupes qui veulent envahir son territoire. Les poulets dominants plutôt que le coq serait constamment à proximité du coq pour chercher de la nourriture tandis que les soumis se détestent à la périphérie du groupe pour chercher de la nourriture. Pour les poussins, ils recherchent de la nourriture autour de leur mère. Considérant les comportements sociaux et les ordres hiérarchiques de poulet comme un essaim, où ils agissent comme un algorithme d'optimisation de l'essaim de poules en équipe individuelle est développé comme suit:

- Dans un essaim de poulets, il y aura plusieurs groupes où chaque groupe aura un coq dominant, un couple de poules et plusieurs poussins. En fonction de la valeur de la fonction objective des poulets, des coqs et d'autres membres, ils seront affectés au sein d'un groupe. Le poulet à la meilleure valeur de la fonction objective sera le coq tandis que le pire serait un poussin. Les poulets restants seront divisés en poules, poussins et poules mères. La relation entre la mère et l'enfant dans un groupe sera choisie au hasard.
- L'ordre hiérarchique, la relation mère-enfant et la relation de dominance dans un groupe seront inchangés pour un pas de temps spécifique (G). Toutes ces caractéristiques mettront à jour leur valeur après les pas de temps.
- Tous les poulets sont guidés par leurs coqs pour chercher de la nourriture alors qu'ils pourraient empêcher les poulets d'autres groupes de voler leur nourriture. Pour les poussins, ils chercheront de la nourriture autour de leur mère et la dominante aurait accès à la nourriture en premier.
- Dans un espace de recherche de dimension D pour CSO, les numéros RN, HN, CN et MN représentent respectivement les coqs, poules, poussins et poules mères. Alors que le meilleur poulet est considéré comme coq et le pire comme poussin, tous les poulets mettront à jour leur position marquée $x_{i,j}^t (i \in [1, \dots, N], j \in [1, \dots, D])$ après un laps de temps ou un pas de temps précis G. Les coqs qui ont la meilleure valeur auront la priorité pour accéder à la nourriture. Pour le rendre simple et réaliste pour le scénario de simulation, la situation est remplacée par la gamme d'allures où le coq a un accès large pour rechercher de la nourriture plutôt que les poulets de valeur de fitness pires (poules, poussins). La position du coq peut être mise à jour par l'équation suivante:

$$x_{i,j}^{t+1} = x_{i,j}^t \times (1 + \text{Random}(0, \delta^2)) \quad (\text{III.6})$$

$$\delta^2 = \begin{cases} 1 & \text{si } f_i \leq f_k \\ \exp\left(\frac{f_k - f_i}{|f_i| + \varepsilon}\right) & \text{avec } k \in [1, N], k \neq i \end{cases} \quad (\text{III.7})$$

Où $\text{Random}(0, \delta^2)$, est une distribution gaussienne avec une moyenne de 0 et un écart type δ^2 , est la plus petite constante de l'ordinateur qui est utilisée pour éviter l'erreur de division

zéro et k est utilisé comme indice de coq sélectionné au hasard parmi les groupes. Ici, f est la valeur de fitness pour le groupe correspondant de x .

- Pour les poules, qui suivent essentiellement le coq du groupe pour chercher de la nourriture et ayant tendance à voler la nourriture d'autres poulets de différents groupes ainsi que d'avoir un ordre de domination, peuvent mettre à jour leur vitesse en utilisant les équations suivantes:

$$x_{i,j}^{t+1} = x_{i,j}^t \times (x_{r1,j}^t - x_{i,j}^t) + S2 \times Rand \times (x_{r2,j}^t - x_{i,j}^t) \quad (III.8)$$

$$S1 = \exp\left(\frac{f_i - f_{r1}}{|f_i| + \varepsilon}\right) \quad (III.9)$$

$$S2 = \exp(f_{r2} - f_i) \quad (III.10)$$

Où $Rand$ est un nombre aléatoire uniforme sur l'intervalle $[0, 1]$, $r1 \in [1, \dots, N]$ est un indice du coq qui est le compagnon de groupe de l' i ème poulet, tandis que $r2 \in [1, \dots, N]$ est un indice du poulet (coq ou poule) qui est choisi aléatoirement dans l'essaim où $r1 \neq r2$. Ici, si $S1 = 0$, alors l' i ème poulet cherchera de la nourriture suivie d'autres poulets et si $S2 = 0$, le poulet ne cherchera pas de la nourriture que sur son propre territoire et ne montrera pas le comportement de vol pour obtenir la nourriture d'autres groupes.

- Pour les poussins, qui se déplacent autour de leur mère et recherchent de la nourriture peuvent mettre à jour leur valeur en utilisant l'équation suivante:

$$x_{i,j}^{t+1} = x_{i,j}^t + FL \times (x_{m,j}^t - x_{i,j}^t) \quad (III.11)$$

Où $x_{m,j}^t$ est la position de l' i ème mère des poussins ($m \in [1, N]$). $FL \in [0, 2]$ considère comme un paramètre qui indique la tendance des poussins à suivre leur mère pour chercher la nourriture [Meng et al, 2014].

Pour obtenir une meilleure compréhension et une vue simple, le pseudo-code de l'algorithme CSO est donné comme ci-dessous:

```

Initialiser une population de N poulets et définir les paramètres associés;
Évaluez par fitness les valeurs de chacun des N poulets,  $t = 0$ ;
Tant que ( $t < \text{Max d'itération}$ )
  Si  $t \% G == 0$  alors
    Classer selon les valeurs de fitness les poulets et établir un ordre hiérarchique
    dans l'essaim;
    Divisez l'essaim en groupes et déterminez la relation entre les poussins et les
    poules mères dans chaque groupe;
  Fin-si.
  Pour  $i = 1 \dots N$ , faire
    Si ( $i == \text{coq}$ )
      Mettre à jour sa solution / son emplacement à l'aide de l'équation (III.6);
    Sinon
      Si ( $i == \text{poule}$ )
        Mettre à jour sa solution /son emplacement à l'aide de l'équation (III.8);
      Sinon
        Si ( $i == \text{poussin}$ )
          Mettre à jour sa solution /son emplacement à l'aide de l'équation
          (III.11);
    Fin Si
  Évaluer la nouvelle solution;
  Si la nouvelle solution est meilleure que la précédente, mettez-la à jour;
Fin Pour i
Fin Tant que
Visualiser les résultats

```

Figure III.12 Pseudo code de l'Algorithme de CSO.

III.3.10 Algorithme d'optimisation par Fruit Fly

III.3.10.1 Aspect biologique :

La drosophile ou mouche des fruits (Fruit Fly en anglais) est un insecte qui préfère de vivre dans des régions tempérées et tropicales, nourrissent principalement de plantes et de fruits en décomposition. Son corps (Figure III.13) contient des organes olfactifs pour permettre de capturer de manière sensible les odeurs sur des dizaines de kilomètres. Pour rechercher de la nourriture, cette insecte utilise un processus repose principalement sur le transfert mutuel d'informations d'odeurs pour effectuer la migration de l'ensemble de la population vers l'insecte à la meilleure odeur.

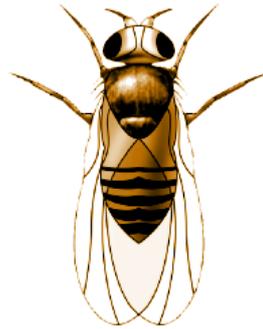


Figure III.13 Structure corporelle de la drosophile

III.3.10.2 Modèle artificielle

En 2011 et dans [Pan, 2011], le professeur Taïwan "Wen-Tsao Pan", simule le comportement de recherche de nourriture par la drosophile comme un modèle d'optimisation appelé FOA. L'optimisation par Fruit Fly est un nouvel algorithme de calcul évolutionnaire inventé pour résoudre les problèmes d'optimisation globale. Par rapport à l'optimisation par des essaims de particules (PSO), l'algorithme génétique (GA) et à d'autres méta-heuristiques classiques d'optimisation, l'algorithme FOA est simple, facile à mettre en œuvre et peut atteindre rapidement une optimisation précise.

Dans le modèle artificiel de l'algorithme, la taille de la population est invariante et pour atteindre l'objectif recherché, le FOA utilise une fonction objective pour estimer la concentration d'odeur. Ensuite, l'essaim vole par vision vers la mouche des fruits la plus proche de la nourriture. À chaque fois, elle obtient le meilleur emplacement de l'insecte où la concentration de l'odeur est la plus élevée. Comme le montre la Figure III.14, cette figure montre un diagramme simplifié de l'évolution de la recherche par la drosophile. La forme ovale dans le schéma représente la zone de recherche dans laquelle un groupe de mouches des fruits est déployé de manière aléatoire pour la traversée de l'espace de la solution. Les points P1, P2 et P3 du schéma indiquent des emplacements candidats obtenus à travers un vol de l'essaim à partir la position initiale, et P2 est le meilleur emplacement sélectionné pour la prochaine itération de recherche. Dans la nouvelle recherche P1', P2' et P3' sont les nouveaux candidats obtenus à partir du centre de l'essaim sur P2. Ce processus se répète jusqu'à ce que l'objectif de la recherche soit trouvé.

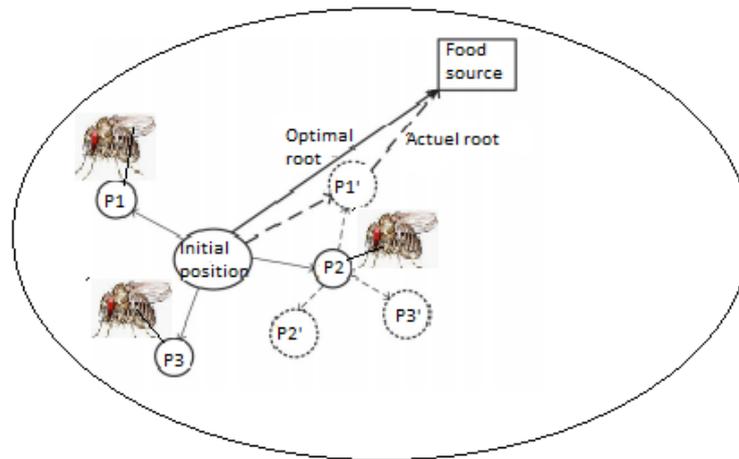


Figure III.14. Schéma de recherche de la drosophile.

Selon les caractéristiques du groupe des mouches des fruits, les étapes de l'algorithme FOA sont décrites comme suite [Pan, 2011]:

1. Initialisez aléatoirement l'emplacement initial du groupe de mouches des fruits, comme illustré sur la Figure III.15.
Init (X_axis, Y_axis);
2. Recherchez la direction de la nourriture en donnant des valeurs aléatoires de la direction et à la distance de chacun des mouche des fruits.
 $X_i = X_axis + \text{RandomValue}();$
 $Y_i = Y_axis + \text{RandomValue}();$
Où, $i = 1 \dots$ Taille de la population.
3. Calculer la valeur de jugement de concentration (S_i) c'est l'inverse de la distance à l'origine ; cela est calculée parce que l'emplacement de la nourriture ne peut être connu.

$$Distance_i = \sqrt{X_i^2 + Y_i^2}; S_i = \frac{1}{Distance_i};$$

4. La valeur de concentration d'odeur (S_i) est substituée dans la fonction de jugement de concentration d'odeur (fonction objective) pour déterminer la concentration d'odeur ($Smell_i$) de la position de l'individu i .
 $Smell_i = \text{Function}(S_i);$
5. Déterminer l'individu avec la concentration optimale (maximale ou minimale selon l'objective d'optimisation) d'odeur parmi l'ensemble des individus de la population.
[bestsmell bestindex]= optimal (smell_i);
6. Conservez la meilleure valeur de concentration de l'odeur et leur coordonnées x, y. les individus de l'essaim vole vers la position optimale par vision.

Smellbest = bestSmell
 X_axis = X (bestIndex)
 Y_axis = Y (bestIndex)

7. Entrez dans une optimisation itérative, en répétez les étapes d'exécution 2 à 5 et jugez si la concentration d'odeur est meilleure que la concentration d'odeur de l'itération précédente, si oui, exécutez l'étape 6.

Les étapes détaillées de l'algorithme FOA sont présentées par l'organigramme de la Figure III.15 suivant :

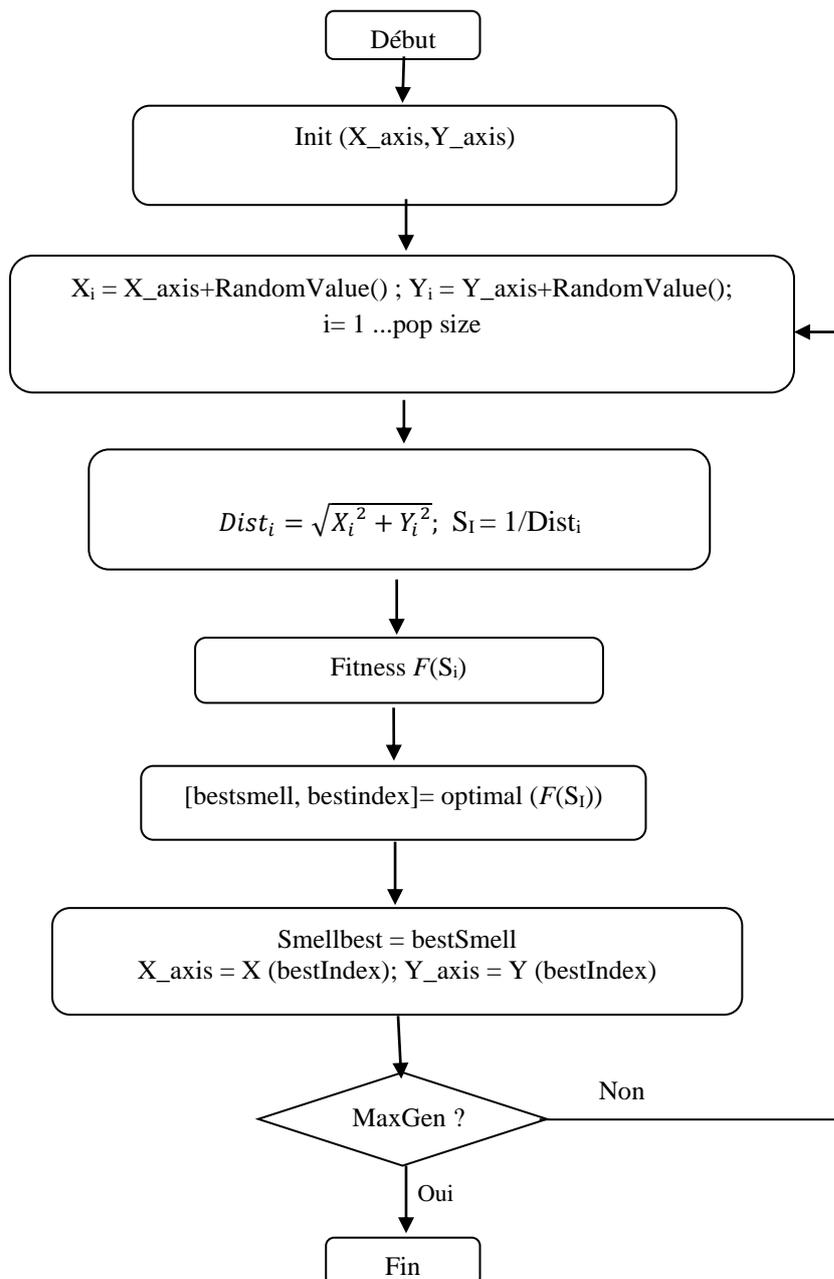


Figure III.15 Organigramme de l'algorithme FOA

III.4. Conclusion

La nature et la biologie sont des nouvelles sources d'inspiration de nombreuses méta-heuristiques. Ainsi, les théories de l'évolution et les phénomènes observés dans les systèmes naturels ont permis l'élaboration d'approches performantes applicables à une grande variété de problèmes d'optimisation difficile. Dans le cadre de notre travail on s'intéresse, plutôt, à l'optimisation de la localisation dans les RCSFs par des méthodes méta-heuristiques. Ces techniques ont prouvé leurs succès dans plusieurs problèmes de divers domaines, et nous essayons de l'adopter au problème de la localisation.

Le chapitre suivant présentera une étude détaillée des différentes contributions dans le cadre des travaux de cette thèse.

Chapitre IV

APPROCHES PROPOSÉES

Chapitre IV

APPROCHES PROPOSÉES

IV.1 Introduction

Dans les chapitres précédents, nous avons expliqué de façon générale le fonctionnement des réseaux de capteurs sans fil, ainsi que ses différents composants, ce qui nous a permis de connaître leurs performances, leurs challenges ainsi que leurs défis. Par la suite, nous avons présenté les différents points qui régissent les systèmes de localisation des nœuds déployés aléatoirement dans la phase de mise en place des capteurs du réseau; plus exactement, nous avons présenté et décrit de façon détaillée les algorithmes de localisation existants. Ces algorithmes, malgré leurs résultats importants, leur précision nécessite une optimisation. Une vue globale d'optimisation et les différentes méta-heuristiques de résolution sont présentées dans le troisième chapitre.

Dans ce chapitre, nous avons mis en œuvre de nouveaux algorithmes distribués en utilisant les méta-heuristiques bio-inspirées. Ils permettent d'apporter des réponses aux nombreuses problématiques posées de la localisation dans les RCSF. En effet, dans ce qui suit, l'algorithme d'optimisation par Fruit Fly (FOA) est adapté pour optimiser des techniques de localisation existantes (DV-HOP et DV-DISTANCE). Par ailleurs, un nouvel algorithme de localisation basé seulement sur la méta-heuristique FOA est proposé. Ainsi, une discussion des différents résultats de simulation obtenus est faite pour chaque contribution.

IV.2. Optimisation des techniques de localisation existantes par FOA

IV.2.1 Travaux similaires

Un aperçu détaillé de la littérature sur les méthodes qui améliorent l'algorithme original DV-Hop (présenté dans le chapitre II) pour la localisation dans les réseaux de capteurs sans fil est effectuée par A.Kaur et al. Dans [Kaur et al, 2017]. Nous résumons les principales approches dans la présente section.

Dans [Chen et al, 2008], H. Chen et al. proposent une variante de DV-Hop rapide, précis et facile à utiliser sous le nom IDVLA, qui améliore la précision ainsi que la couverture. Ils ont observé que si les ancres sont déployées uniformément, les performances de l'algorithme IDVLA augmentent. Les résultats expérimentaux montrent qu'avec 10% des ancres, l'algorithme IDVLA atteint une couverture de localisation d'environ 100% et la précision est améliorée de 9% avec 5% des ancres [Chen et al, 2008]. L'algorithme de localisation centroïde pondéré abrégé WCL a été proposé dans [Zhang et al, 2012] pour améliorer la complexité de calcul de DV-Hop. Les résultats expérimentaux montrent que les performances de l'algorithme WCL sont presque les mêmes que celles de l'algorithme DV-Hop original en termes de précision. Cependant, la complexité de calcul de l'algorithme WCL est inférieure à celle de l'algorithme DV-Hop, car elle contient une phase de moins que l'algorithme DV-Hop original.

Dans [Zhang et al, 2012], B.Zhang et al. ont proposé un autre algorithme de centroïde pondéré amélioré basé sur DV-Hop appelé IWCL pour améliorer la précision. Les résultats de la simulation montrent que l'algorithme IWCL fonctionne mieux que l'algorithme DV-Hop en termes de précision. L'avantage de l'algorithme IWCL est qu'il est plus précis et moins complexe. L'inconvénient est que le nombre d'ancres qui doit être sélectionné pour calculer l'emplacement d'un nœud inconnu n'est pas mentionné dans la phase 1.

Dans [Yu et Li, 2012], W.Yu et al ont proposé IDV-Hop, c'est une version de localisation amélioré de DV-Hop qui ajoute une étape de correction tout en calculant la distance entre un nœud inconnu et l'ancre dans la deuxième phase afin d'améliorer la précision. Comme dans cet algorithme, une étape de correction est rajoutée, ce qui augmente la complexité de calcul de l'algorithme donné dans une certaine mesure, bien que la précision soit améliorée par rapport à l'algorithme DV-Hop original. Les résultats de la simulation montrent que la précision est améliorée de 1,5% à 4% par rapport à l'algorithme DV-Hop d'origine en faisant varier le nombre des ancres.

H.Safa et al. [Safa, 2014] a proposé une version hybride sous le nom HDV-Hop pour minimiser les erreurs de localisation, les inondations et la consommation d'énergie de DV-Hop. Dans cet algorithme hybride, les auteurs ont supposé que des ancres étaient déployées sur le périmètre du réseau. L'avantage de l'algorithme HDV-Hop est qu'il réduit les inondations. Les ancres et le nœud inconnu doivent envoyer le message aux ancres qui

sont connectés à la station de base. La plupart des calculs sont effectués par la station de base. La simulation et les résultats dans [Safa, 2014] montrent que cet algorithme réduit l'erreur de localisation et la consommation d'énergie par rapport à l'algorithme DV-Hop original. L'inconvénient de cet algorithme est que les ancres et la station de base de cet algorithme doivent être déployées en dehors du réseau. Si la station de base échoue, cet algorithme ne fonctionnera pas. Ainsi que, un protocole de routage est requis pour obtenir le chemin le plus court du nœud inconnu à l'ancre parent et pour obtenir le chemin de l'ancre à la station de base.

Bo.Peng et al [Peng et Li, 2015] ont proposé un algorithme de localisation basé le DV-Hop et la méta-heuristique de l'algorithme génétique, appelé algorithme GADV-Hop. GADV-Hop utilise l'algorithme génétique pour améliorer la précision de l'algorithme DV-Hop traditionnel. L'avantage de cet algorithme est qu'il fonctionne mieux que l'algorithme DV-Hop original car il entraîne une erreur de localisation plus faible et il est beaucoup plus stable. Cependant, l'inconvénient de cet algorithme est qu'il augmente sa complexité de calcul en raison de l'algorithme génétique.

Dans [Song et Tam, 2015], G.Song et al. ont proposé l'algorithme Hyperbolic DV-Hop utilisé pour améliorer la précision. Par rapport à l'algorithme DV-Hop d'origine, l'algorithme Hyperbolic DV-Hop permet d'améliorer la précision de 8 à 10%. L'algorithme Hyperbolic-DV-Hop améliore la précision dans une faible mesure, ainsi dans [Song et Tam, 2015], G.Song et al. ont proposé un deuxième algorithme de localisation appelé algorithme de localisation DV-Hop centroïde pondéré amélioré abrégé IWCDV-Hop. Cet algorithme utilise un concept de centroïde pour améliorer la précision dans une large mesure. Cet algorithme utilise tous les ancres du réseau pour calculer l'emplacement d'un nœud inconnu, ce qui améliore sa précision. Comparé au DV-Hop original, le « IWCDV-Hop » augmente sa précision de 59% à 63%.

Dans [Gui et al, 2015], L.Gui et al ont proposé l'algorithme de localisation CheckOut DV-Hop dans lequel une phase supplémentaire appelée phase de paiement est ajoutée pour améliorer la précision. Dans cet algorithme, le nœud inconnu utilise l'ancre la plus proche pour calculer son emplacement. La complexité de calcul est presque la même que celle de l'algorithme DV-Hop original. L'algorithme de localisation DV-Hop Checkout améliore la précision de 10 à 25% par rapport à l'algorithme DV-Hop d'origine. L.Gui et al. [Gui et al, 2015] ont proposé une autre variante de DV-Hop. Cette version sélectionne

seulement 3 ancrés et les utilise pour calculer les coordonnées d'un nœud inconnu. Les résultats expérimentaux montrent que l'utilisation de cet algorithme améliore la précision de 18 à 30% dans différents scénarios par rapport à l'algorithme DV-Hop original. La complexité de calcul augmente dans cet algorithme en raison de la troisième étape qui nécessite une procédure très complexe et longue pour trouver les trois ancrés les plus appropriés au nœud inconnu.

Dans [Tomic et Mezei, 2016], S. Tomic et al. ont proposé un algorithme DV-Hop amélioré, appelé algorithme iDV-Hop1, pour améliorer la précision dans tous les types de scénarios en utilisant la géométrie. Les résultats de la simulation montrent que l'algorithme iDV-Hop1 améliore la précision jusqu'à trois fois dans des scénarios à topologie irrégulière par rapport à l'algorithme DV-Hop original. Cependant, l'algorithme iDV-Hop1 a augmenté la complexité de calcul par rapport à l'algorithme DV-Hop original en raison de l'ajout d'une quatrième phase contenant la géométrie.

Dans [Tomic et Mezei, 2016], S. Tomic et al. proposent une deuxième amélioration (iDV-Hop2), basée sur la géométrie pour améliorer la précision qu'utilise l'algorithme DV-Hop original pour tous les scénarios et avec iDV-Hop1 pour les scénarios avec des topologies régulières. Les résultats de la simulation prouvent que l'algorithme iDV-Hop 2 a jusqu'à 11% de précision inférieure à l'algorithme iDV-Hop 1 et l'algorithme DV-Hop original. Dans [Tomic et Mezei, 2016], S.Tomic a proposé un troisième algorithme appelé Quad DV-Hop pour réduire l'erreur de localisation, dans lequel un raffinement dans la troisième étape de l'algorithme DV-Hop original est effectué. Les résultats de la simulation prouvent que l'algorithme Quad DV-Hop a donné de meilleures performances dans tous les types de scénarios par rapport à l'algorithme DV-Hop original, mais il a une complexité de calcul élevée en raison de l'utilisation du solveur QP.

IV.2.2 Formulation de Problème

Dans le réseau de capteurs sans fil, nous supposons qu'il y a N capteurs déployés aléatoirement sur un espace de dimension D ($D=2$ dans notre travail). L'objectif de la localisation est d'estimer les coordonnées de $N-M$ nœuds inconnus à l'aide d'informations a priori sur les positions de M ancrés. La distance entre la position estimée et la position réelle des nœuds qui représente l'erreur de la localisation est calculées par l'équation suivante:

$$Error_i = \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2} \quad (IV.1)$$

Où (\hat{x}_i, \hat{y}_i) les coordonnées estimées du capteur i et (x_i, y_i) ses coordonnées réelles. $i = (1, 2 \dots N-M)$

Le critère le plus important du système de localisation est l'erreur moyenne. Dans ce travail, il est calculé par la formule suivante:

$$Error_{ave} = \frac{1}{M} * \sum_{i=1}^M Error_i \quad (IV.2)$$

La fonction objective de l'algorithme FOA pour la localisation des nœuds peut être calculée comme suit:

$$f(x, y) = \frac{1}{M} * \sum_{i=1}^M (\sqrt{(x - x_i)^2 + (y - y_i)^2} - s_i)^2 \quad (IV.3)$$

Où

- $M \geq 3$ est le nombre de nœuds d'ancrage.
- (x_i, y_i) est les coordonnées du i ème ancre
- (x, y) sont les coordonnées de l'individu à évaluer.

S_i est la somme de la racine carrée de distances estimé entre le nœud recherché et les ancres à visibilité directe avec ce nœud.

IV.2.3 DV-HOP(FOA) : Optimisation de DV-HOP par FOA

IV.2.3.1 Présentation

Afin d'optimiser la précision de la localisation de l'approche traditionnelle DV-HOP (présenté dans le chapitre II), cet algorithme proposé exécute d'abord l'algorithme de localisation DV-HOP pour obtenir les emplacements initiaux des nœuds. Par la suite, l'algorithme DV-HOP(FOA) initialise un essaim des Fruit Fly dans ces emplacements initiaux obtenus par DV-HOP. Le processus de localisation est divisé en trois phases:

1. **Initialisation:** N capteurs de rayon R sont déployés aléatoirement avec M parmi eux sont des ancres dans une zone bidimensionnelle; ensuite, nous déterminons le nombre des individus ($SizPop$) et le nombre maximale des générations ($MaxGen$) utilisées pour l'algorithme d'optimisation par Fruit Fly (FOA).
2. **Recherche initiale:** dans cette deuxième phase, nous utilisons l'algorithme de localisation classique DV-HOP pour estimer les emplacements initiaux des

différents nœuds inconnus. Dans cet algorithme classique, chaque ancre calcule et diffuse la distance moyenne par saut pour tous les nœuds du réseau, et chaque nœud inconnu est localisé par la multilatération en se basant sur les positions des ancres. Les positions obtenues seront considérées comme des emplacements initiaux pour les différents nœuds inconnus.

3. **Optimisation:** dans cette phase, nous appliquons l'algorithme d'optimisation par Fruits Fly (FOA) pour minimiser l'erreur de localisation obtenue par DV-HOP afin d'obtenir les coordonnées relatives des nœuds inconnus comme suit:
 - Pour chaque nœud inconnu, nous plaçons un essaim de N' ($N' = \text{SizPop}$) individus dans leur emplacement initial estimé par DV-HOP;
 - Donner une valeur aléatoire de la direction et de la distance de chaque individu ; découvre l'individu avec la valeur d'odeur la plus basse en utilisant la fonction objective et en gardant ses coordonnées;
 - Répéter le processus en comparant les positions précédentes avec les positions actuelles, en gardant les meilleures en termes de fonction objective jusqu'à ce que la condition d'arrêt (MaxGen) soit terminée.
 - Considérez les dernières coordonnées obtenues comme une position relative à ce nœud inconnu.

Les étapes détaillées de l'approche DV-Hop(FOA) proposée sont présentées dans l'organigramme de la Figure (IV.1).

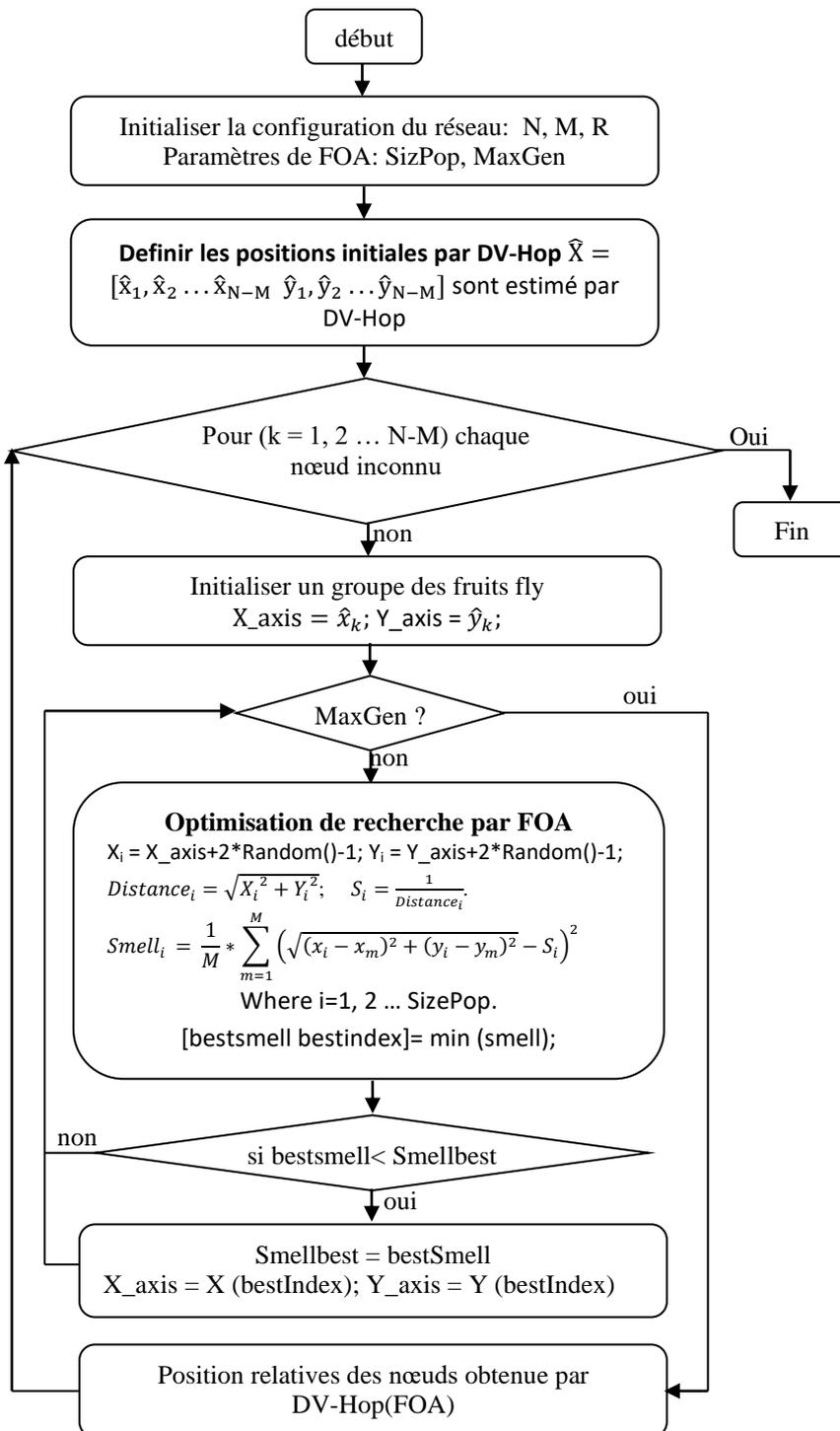


Figure IV.1 Organigramme de l'algorithme DV-Hop(FOA).

IV.2.3.2 Résultats de simulation de DV-Hop(FOA)

Dans cette section, nous utilisons des simulations informatiques sur la plate-forme MATLAB pour évaluer et démontrer les performances de l'algorithme DV-Hop(FOA). Dans toutes nos simulations, nous avons utilisé un environnement matériel présentant les

caractéristiques suivantes : un microprocesseur Intel CoreTMi3 avec 4 Go de RAM, dans un système d'exploitation Windows 7.

Ensuite, nous comparons l'algorithme proposé avec l'algorithme DV-Hop original en termes d'erreur de localisation moyenne calculée par l'équation (IV.1).

La zone d'expérimentation est une topologie sous la forme d'un rectangle carrée avec une taille fixe de 2500m^2 et tous les capteurs du réseau ont un rayon de transmission similaire (R) fixé à 20m. La Figure IV.2 suivante, montre un exemple simple d'un réseau avec $N = 50$ nœuds répartis de manière aléatoire où $M = 10$ sont des ancres. Les points noirs représentent des nœuds avec des positions inconnues et les points rouges représentent les nœuds ancrés. Les paramètres de l'algorithme FOA sont: la taille de la population qui est de 20 individus (mouches des fruits) et le nombre maximal des générations qui est égal à 100 itérations.

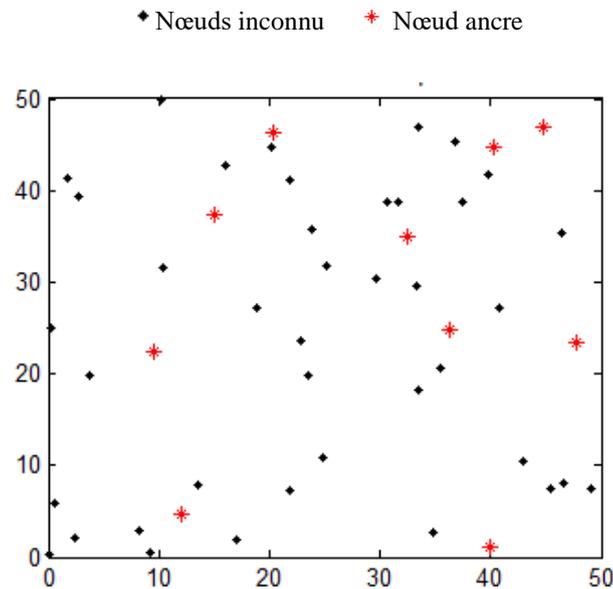


Figure IV.2 Exemple de simulation d'un RCSF avec 50 nœuds placés aléatoirement dans un champ 2D.

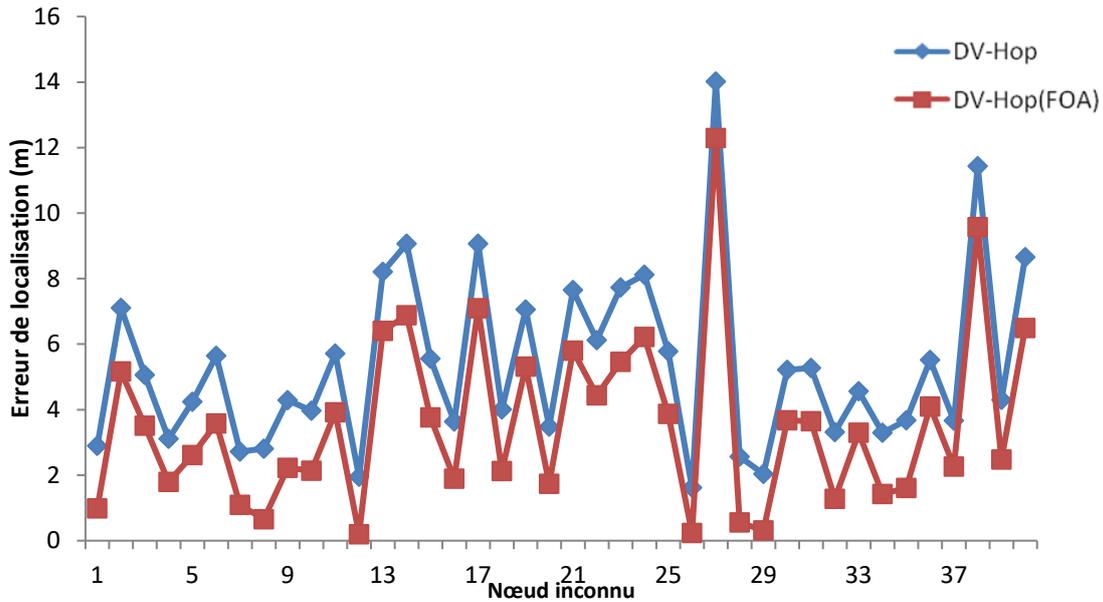


Figure IV.3 Erreur de localisation pour chaque nœud inconnu.

La Figure (IV.3) montre le graphique de l'erreur commise dans la localisation pour chaque nœud inconnu par la méthode classique DV-HOP et l'algorithme DV-Hop(FOA). Ces diagrammes correspondent à la configuration de la Figure IV.2. Ces résultats montrent clairement que l'approche DV-Hop(FOA) proposée minimise l'erreur de localisation dans chaque nœud inconnu obtenue par DV-Hop. Par conséquent, le DV-HOP(FOA) minimise la moyenne d'erreur de localisation (calculée par l'équation (IV.2)) trouvée par le DV-HOP traditionnel, comme nous indique le tableau suivant:

Tableau IV.1 Moyenne d'erreur de DV-Hop (FOA) et DV-Hop

	DV-Hop	DV-Hop(FOA)
Moyenne d'erreur de localisation	5.35206	3.5504

Afin de montrer l'effet du changement des paramètres, nous analysons chaque caractéristique indépendamment. Tout d'abord, nous avons fixé tous les paramètres des algorithmes et nous avons fait varier le nombre de nœuds déployés dans le champ 2-D de 10 à 100. Les résultats obtenus sont présentés sur la Figure (IV.4).

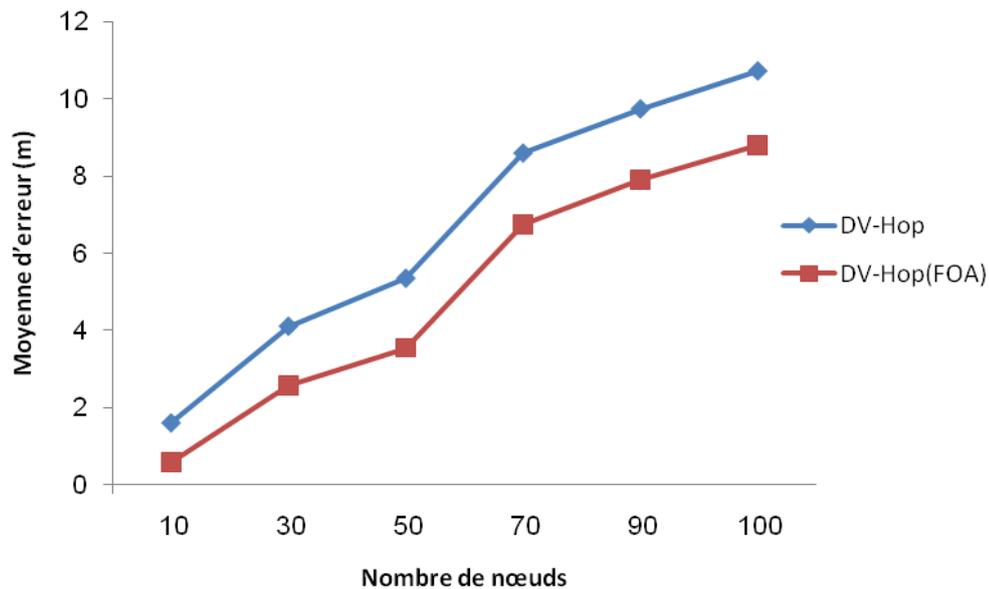


Figure IV.4 Moyenne d'erreur par rapport à la variation du nombre de nœuds.

Deuxièmement, nous déployons de façon aléatoire 50 capteurs dans l'espace 2D. Puis, nous changeons le nombre de nœuds ancrés entre 5 et 20 nœuds et nous fixons tous les autres paramètres pour les deux algorithmes. La Figure (IV.6) illustre les résultats obtenus.

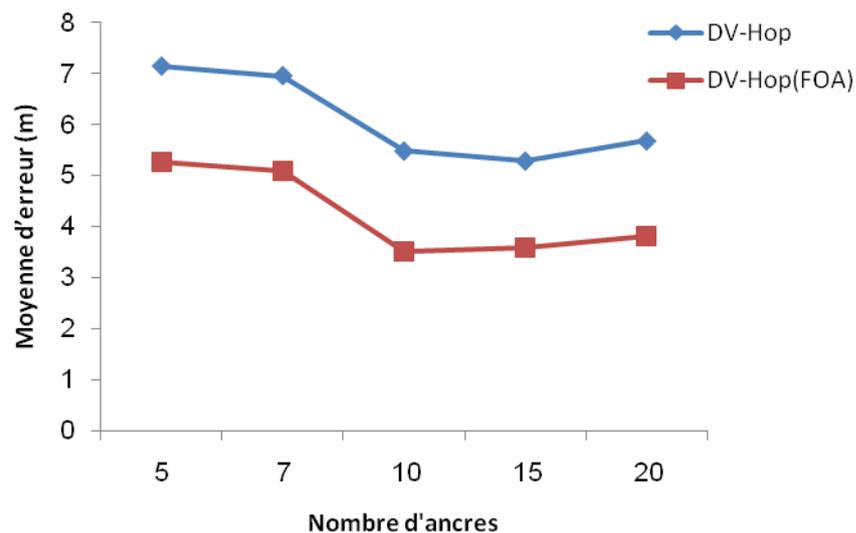


Figure IV.5 Moyenne d'erreur de localisation par rapport à la variation de nombre d'ancres.

Les résultats expérimentaux obtenus sur les Figures (IV.4) et (IV.5) montrent pour les deux algorithmes que l'erreur moyenne estimée augmente lorsque le nombre de capteurs augmente et diminue lorsque le nombre d'ancres augmente jusqu'à 10 ancres.

Ensuite, il se stabilise et augmente légèrement jusqu'à 20 ancrés. Par conséquent, afin de déterminer l'emplacement de tout capteur dont nous avons besoin, au moins, on doit avoir la distance entre ce dernier et trois ancrés. Il est donc probable que nous ne pouvons pas obtenir les bons emplacements de certains capteurs lorsque nous utilisons un nombre insuffisant des ancrés.

Enfin, nous étudions l'effet du changement du nombre des individus sur les résultats obtenus par la méthode DV-HOP(FOA). Pour cela, nous utilisons les paramètres ayant donné les meilleurs résultats dans l'étude précédente et nous varions le nombre de mouches de 5 à 40. Les résultats obtenus montrent l'effet de l'augmentation du nombre des individus (mouches) sur l'erreur moyenne. La Figure IV.6 montre que l'utilisation de 20 individus garantit presque un bon résultat. De plus, lorsque nous obtiendrons un nombre suffisant d'individus proportionnels à la configuration du réseau, nous obtiendrons presque la même précision.

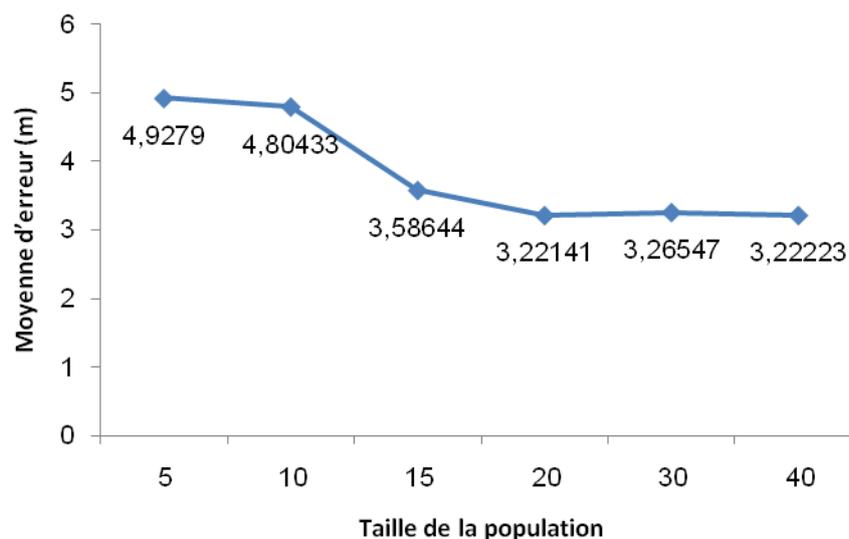


Figure IV.6 Moyenne d'erreur de localisation par rapport à la variation de la taille de la population.

IV.2.3.3 Récapitulatif

Dans cette proposition, l'algorithme d'optimisation par Fruits Fly est utilisé pour minimiser l'erreur de localisation entre les coordonnées réelles et celle estimées pour les différents capteurs inconnus. Dans l'approche proposée DV-Hop(FOA), nous initialisons un groupe de mouches des fruits dans les emplacements initiaux calculés par la méthode DV-Hop dans une étape préliminaire. Les résultats de la simulation montrent que

l'algorithme proposé fournit une erreur de localisation moyenne inférieure (avec un taux de 40% d'amélioration) par rapport à l'algorithme DV-Hop traditionnelle.

Dans cette proposition, nous avons observé la capacité de la méta-heuristique FOA à améliorer la précision de la localisation dans la technique classique DV-HOP. Par conséquent, nous recommandons d'hybrider ces méthodes afin d'obtenir de meilleurs résultats et ces nouvelles méthodes sont moins compliquées que les méthodes traditionnelles et consomment donc moins d'énergie. Il est également nécessaire d'améliorer la précision des méthodes de localisation.

La technique DV-Distance est similaire à DV-Hop dans la mesure où certains chercheurs la classent comme une variante de DV-Hop. La seule différence est qu'en DV-Distance, les distances entre les capteurs sont mesurées en utilisant la puissance du signal RSSI plutôt que le nombre de sauts en DV-Hop. Cependant, DV-Hop ne fonctionnera bien que dans les réseaux isotropes. En revanche, la technique DV-Distance fonctionne bien dans les réseaux non isotropes [WANG et al, 2010]. Dans ce contexte, nous proposons dans la prochaine section l'utilisation de DV-Distance dans la première phase d'un schéma de localisation similaire que DV-Hop(FOA).

IV.2.4 DV-Distance(FOA) : Optimisation de DV-Distance par FOA

IV.2.4.1 Présentation

Dans la présente contribution, nous reprenons l'idée précédente, afin d'améliorer la précision de la localisation obtenue par l'algorithme de localisation DV-Distance. En se basant sur l'algorithme DV-Distance et la méta-heuristique d'optimisation Fruit fly, nous proposons une approche de localisation contenant deux phases principales. Dans la première phase, une solution auxiliaire (position initiales) est produite par l'exécution de la technique DV-Distance. La deuxième phase permet d'affiner les positions initiales. En effet, la solution de la première phase est modifiée en appliquant l'algorithme d'optimisation du Fruit Fly.

Pratiquement, on commence par une étape d'initialisation où N capteurs avec un rayon de transmission similaire R sont déployés de manière aléatoire dans un champ 2D avec $M < N$ nœuds sont considéré comme des ancres. De plus, nous déterminons SizePop (nombre

d'individus) et MaxGen (génération maximale) utilisés pour l'exécution de FOA. Dans ce contexte, le processus de localisation proposé est divisé en deux phases:

Phase 1 (Recherche initiale): dans un premier temps, nous utilisons la méthode DV-Distance (présentée dans le chapitre II) pour estimer une solution auxiliaire de tous les capteurs avec des positions inconnues. A cet effet, chaque ancre calcule et diffuse le rapport de correction permettant aux capteurs inconnus du RCSF de calculer la distance cumulée à cette ancre. Ensuite, chaque nœud inconnu est localisé par trilatération grâce aux informations reçues de trois ancres ou plus. Les positions obtenues seront considérées comme une solution initiale des nœuds inconnus.

Phase 2 (Raffinement): Dans cette deuxième phase, nous utilisons les étapes de FOA afin de réduire l'erreur de localisation tout en déterminant les positions relatives aux nœuds inconnus comme suit:

1. Pour chaque nœud inconnu, nous mettons un essaim avec un total de N' ($N' = \text{SizePop}$) individus dans la position initiale correspondante estimée par l'algorithme DV-Distance et exécuté dans la première phase;
2. Nous modifions grâce à une valeur aléatoire les coordonnées de chaque individu appartenant à l'essaim; nous évaluons ensuite la nouvelle position obtenue en utilisant la fonction objective (équation (IV.3)) afin de calculer la valeur de concentration d'odeur correspondante;
3. Nous découvrons l'individu avec la valeur d'odeur la plus basse et nous considérons son emplacement comme la position de recherche initiale de la prochaine itération. Ensuite, toutes les individus de l'essaim s'envolent vers cette position;
4. Nous répétons les étapes 2 et 3 du processus en comparant l'odeur de la position précédente avec la position actuelle, puis en conservant la meilleure jusqu'à ce que la condition d'arrêt (MaxGen) soit épuisée.
5. Considérez les dernières coordonnées obtenues comme l'emplacement relatif de ce capteur inconnu.

IV.2.4.2 Résultats de simulation de DV-Distance(FOA)

Les scénarios de simulation que nous avons menés pour évaluer les performances de notre schéma de localisation DV-Distance(FOA) se concentrent principalement sur les paramètres de performances suivants:

- **Évolutivité:** pour aborder l'efficacité de la méthode proposée tout en augmentant la densité d'un RCSF.
- **Le taux d'erreur de localisation (LER):** pour évaluer la précision fournie par la technique proposée selon l'équation (IV.4) permettant de calculer le rapport moyen des distances euclidiennes entre les positions réelles (x_i, y_i) des N nœuds inconnus et celles estimées (\hat{x}_i, \hat{y}_i) par notre technique DV-Distance (FOA). Notez que R est le rayon de communication des capteurs:

$$LER = \frac{\sum_{i=1}^N \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}}{N \times R} \times 100\% \quad (IV.4)$$

Dans ce qui suit dans cette section, nous comparons notre méthode DV-Distance(FOA) avec les techniques DV-Hop et DV-Distance traditionnelles ainsi que notre première contribution DV-Hop(FOA). Les paramètres de l'algorithme FOA dans tous les scénarios de simulation que nous avons réalisés sont fixés à 100 itérations pour MaxGen et 20 mouches pour le nombre d'individus (i.e. la valeur de SizePop).

La Figure (IV.7) montre les résultats de simulation obtenus avec les techniques mentionnées précédemment en termes de taux d'erreur de localisation (LER) tout en faisant varier la taille de la zone du RCSF entre 400m² et 10000m². Le nombre total de capteurs est fixé à 50, dont 10 sont des ancres, avec un rayon de communication similaire de 25m.

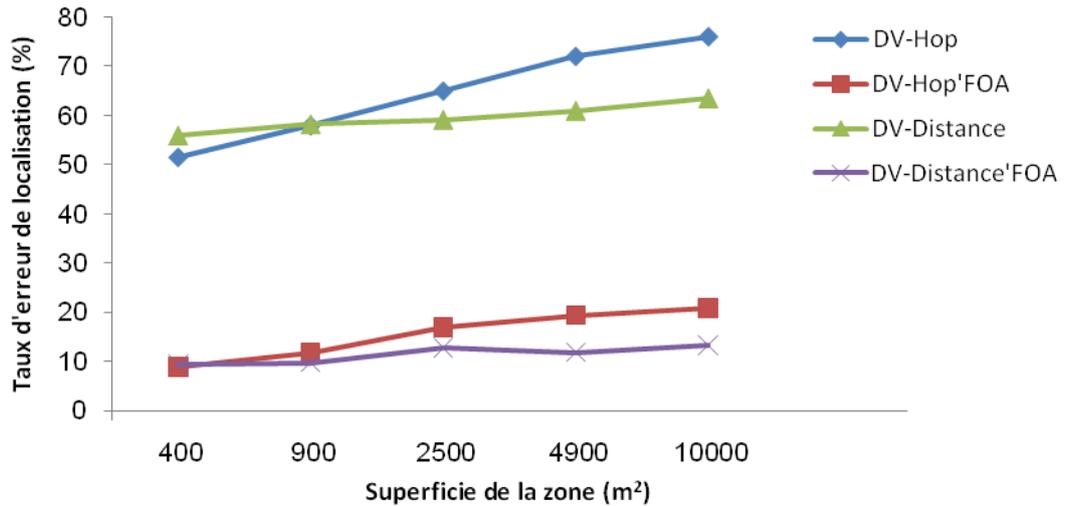


Figure. IV.7 Taux d'erreur de localisation par rapport à la superficie de la zone du RCSF.

Les résultats obtenus illustrés par la Figure (IV.7) montrent que le taux d'erreur de localisation (LER) augmente lorsque la taille de la zone augmente pour tous les algorithmes utilisés dans cette expérience. De plus, les résultats obtenus illustrés sur la Figure (IV.7) montrent la grande capacité des deux approches hybrides par la méta-heuristique FOA à minimiser le taux d'erreur de localisation par rapport aux méthodes traditionnelles. Cette amélioration du LER atteint 43% dans DV-Distance(FOA) avec une légère supériorité par rapport à DV-Hop(FOA).

Dans une deuxième expérience, nous avons étudié l'impact du rayon de communication sur le taux d'erreur de localisation calculé selon l'équation (IV.4), dans un réseau de capteurs sans fil composé de 50 nœuds en utilisant 20% d'ancres déployées dans une zone de 50m*50m. La Figure (IV.8) illustre les résultats obtenus correspondant aux différents algorithmes, tout en faisant varier le rayon de communication de 15m à 40m. Ainsi, ces résultats montrent l'importance du rayon de communication sur le processus de localisation. En effet, plus le rayon augmente, plus l'erreur diminue, pour les quatre algorithmes. Néanmoins, notre méthode DV-Distance(FOA) fonctionne mieux en termes de LER par rapport aux autres algorithmes considérés pour toutes les valeurs du rayon de communication.

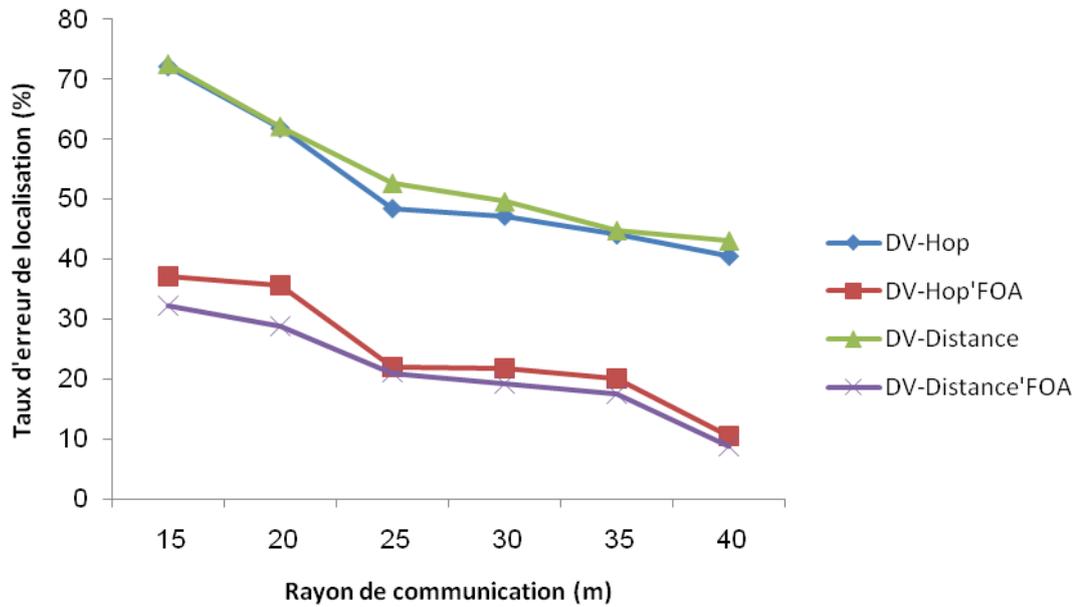


Figure. IV.8 Taux d’erreur de localisation par rapport au rayon de communication.

Dans la dernière expérience, nous avons calculé le taux d'erreur de localisation en variant le pourcentage des ancres utilisés. Les résultats obtenus sont illustrés sur la Figure (IV.9). En plus, du fait que les résultats de cette expérience confirment nos conclusions précédentes concernant la capacité de FOA à apporter une amélioration importante aux algorithmes traditionnels, nous notons également que nous obtenons une meilleure précision en augmentant le nombre d'ancres. Cette amélioration est très importante pour les techniques basées sur le FOA, qui dépend du nombre d'ancres et en particulier avec cette proposition DV-Distance(FOA).

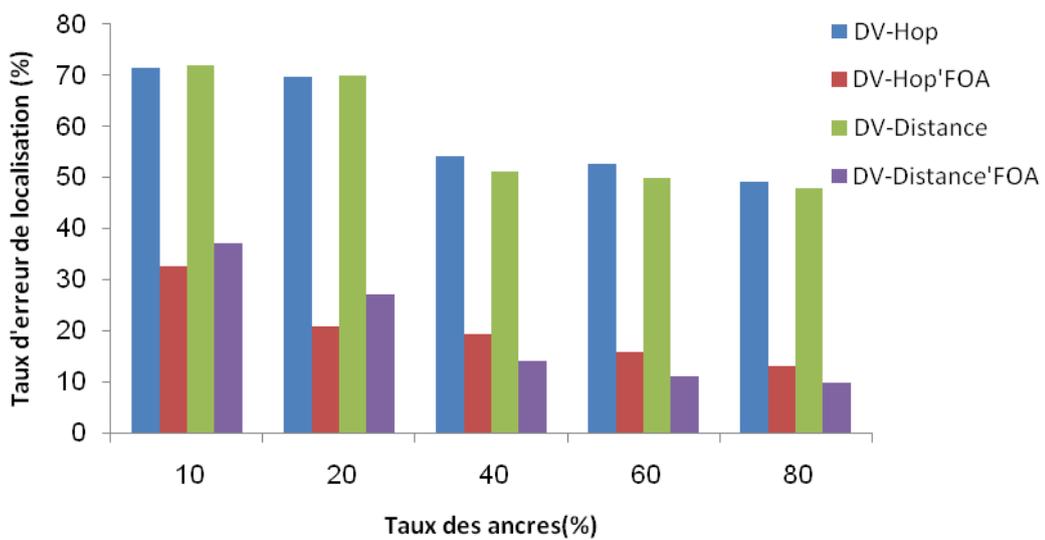


Figure. IV.9 Taux d'erreur de localisation par rapport au Taux des ancres.

IV.2.4.3 Récapitulatif

Sur la base de la méta-heuristique de FOA ainsi que de l'algorithme de localisation DV-Distance, nous avons proposé une nouvelle approche de localisation en deux phases pour RCSFs. Cette technique de localisation hybride repose sur deux phases principales; une phase permettant d'estimer les positions préliminaires déterminées par DV-Distance et une autre phase de raffinement FOA permettant de réduire le taux d'erreur de localisation. Les expériences de simulation que nous avons menées ont validé l'effet de l'algorithme proposé pour réduire 43% d'erreur de localisation par rapport à la technique traditionnelle de localisation DV-Distance. D'un autre côté, la méthode DV-Distance(FOA) fonctionne légèrement mieux que la technique DV-Hop(FOA).

Ce succès de FOA d'optimiser les techniques classiques, telles que DV-Distance et DV-HOP, nous a encouragé à proposer une approche de localisation basée seulement sur cette méta-heuristique prometteuse, qui sera l'objet de la prochaine section.

IV.3 FOA-L : Une nouvelle méta-heuristique de localisation

IV.3.1 Travaux similaires

Récemment, de nombreuses méta-heuristiques ont été utilisées pour améliorer la précision des algorithmes de localisation dans les réseaux de capteurs sans fil. Cette section présente brièvement plusieurs travaux publiés dans la littérature dans ce contexte.

Les auteurs de [Kannan et al, 2006], appelant la méta-heuristique du recuit simulé pour produire une technique de localisation bio-inspirée qu'ils l'ont appelé SAL. L'algorithme SAL identifie l'emplacement des nœuds déployé de manière aléatoire dans une zone d'intérêt. Le résultat de simulation obtenu par KANNAN et al. en 2006 [Kannan et al, 2006], a montré que SAL est meilleure que la technique de localisation basée sur la programmation semi-définie présentée par Biswas et Ye en 2004 [Biswas et Ye, 2004] en terme de précision de la localisation. Mais, comme la plupart des techniques d'optimisation existantes sur cette période, la méthode SAL souffre également du problème du minimum local.

Dans [Marks et Niewiadomska, 2007], les auteurs ont proposé une approche de localisation en deux phases qui hybride l'algorithme génétique dans la première phase avec

l'algorithme de recuit simulé dans la deuxième phase. Ce schéma hybride basé sur la bio-inspiration permet de traiter le problème d'ambiguïté de la localisation.

Gopakumar et Jacob en 2008 ont répondu à la limite principale de SAL [Gopakumar et Jacob, 2008]. Ils se sont basés sur la méta-heuristique de l'optimisation des essaims de particules afin de diversifier la recherche dans la zone d'intérêt pour trouver les positions des nœuds cibles. L'idée principale de la localisation par essaims de particules réside dans le placement aléatoire d'un groupe d'essaims dans toute la zone de recherche et évaluer les différents emplacements des particules par une fonction objective. Chaque position de particule représente un optimum local au nœud cible. La position avec la meilleure valeur de fitness est considérée comme un optimum global. Ensuite, l'algorithme répète ce processus en mettant à jour la vitesse et la position de chaque itération jusqu'à un nombre maximum d'itérations. Les résultats de simulation de comparaison entre les performances de PSO et celles de SAL pour la localisation des capteurs ont montré que PSO a de meilleures performances que SAL. Cependant, le principal inconvénient de PSO est la nécessité d'un grand nombre d'ancres pour localiser tous les capteurs inconnus.

Une autre méta-heuristique de localisation basée sur les principes de sélection, de croisement et de mutation de l'algorithme génétique est présentée dans [Nan et al, 2007]. Cette méthode définit les positions de capteurs inconnus en calculant leurs distances par rapport aux ancres voisins à un seul saut. Le résultat de la comparaison avec les travaux antérieurs a montré que l'algorithme génétique utilise moins d'ancres. Récemment, de nouvelles méthodes bio-inspirées sont utilisées pour la localisation dans RCSF. Parmi ces méthodes évolutives, nous avons: l'optimisation par colonies de fourmis [Sivakumar et Venkatesan, 2016], l'algorithme d'optimisation des abeilles [Moussa et El-Sheimy, 2010], l'algorithme Firefly [Cao et al, 2012], l'algorithme de recherche coucou [Goyal et Patterh, 2014] et l'algorithme d'essaim de poulet [Al Shayokh et Soo, 2017].

Dans cette thèse, nous proposons une nouvelle approche intelligente basée sur l'algorithme d'optimisation par Fruits Fly (FOA). Ainsi, nous utilisons le comportement de recherche de nourriture dans la méta-heuristique FOA pour rechercher les coordonnées des nœuds cibles. L'approche bio-inspirée du FOA est utilisée car elle a été récemment inventée pour trouver des emplacements dans un espace de recherche multidimensionnel et bénéficier des avantages de la méta-heuristique afin d'améliorer la précision de localisation dans les RCSFs.

IV.3.2 Présentation de FOA-L

Dans cette approche, nous avons adapté l'algorithme d'optimisation par Fruit Fly pour produire un nouvel algorithme de localisation appelé FOA-L. Pour calculer les coordonnées de N capteurs inconnus, le processus de notre approche proposée FOA-L est le suivant:

1. Un nombre total de M ancrs et N capteurs inconnus sont déployés aléatoirement dans un espace bidimensionnel. Tous les capteurs N+M (inconnus ou ancrs) ont un rayon de transmission R similaire. Les capteurs inconnus, qui estiment leur position à la fin d'une itération, servent des ancrs lors de la prochaine itération et se comportent comme des ancrs estimées.
2. Chaque ancre diffuse un message d'ancrage contenant son identifiant, ses coordonnées et des informations de signal permettant de mesurer le RSSI correspondant.
3. Les capteurs inconnus qui sont dans le rayon de transmission de moins de trois ancrs sont considérés comme des nœuds non localisés ; donc, ils attendent la prochaine itération pour se localiser. D'autre part, lorsqu'un capteur avec une position inconnue reçoit trois messages ou plus de trois ou plusieurs ancrs différentes (y compris les ancrs estimées), alors il est considéré comme un capteur localisable et il démarre notre processus proposé permettant d'estimer ses coordonnées.
4. Chaque capteur considéré comme un nœud localisable, utilise les informations RSSI pour mesurer la distance entre lui et les ancrs avec un seul saut. Les mesures RSSI sont dégradées avec du bruit gaussien n_i , en raison d'obstacles liés à l'environnement. Un capteur cible estime sa distance à partir de la $i^{\text{ème}}$ ancre comme $\hat{d}_i = d_i + n_i$ où d_i est la distance réelle donnée par l'équation (IV.4)

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (\text{IV.4})$$

Où, (x, y) sont les coordonnées du capteur inconnu et (x_i, y_i) sont les coordonnées de la $i^{\text{ème}}$ ancre au voisinage du capteur inconnu. Il est entendu que le bruit de mesure n_i a une valeur aléatoire uniformément répartie dans la plage $d_i \pm d_i \frac{p_n}{100}$ où P_n est le pourcentage de bruit dans la mesure de distance [Cho et al, 2011].

5. Chaque capteur localisable exécute les étapes suivantes de FOA-L proposée afin de se localiser (voir la Figure IV.10 pour une illustration d'un cas d'utilisation simplifié de notre méthode proposée avec trois ancres et trois mouches des fruits):

5.1. Le nombre de mouches des fruits (SizePop) et le nombre maximal des générations (MaxGen) que nous utilisons pour l'exécution de l'algorithme FOA dans le contexte de la localisation sont initialement prédéterminés.

5.2. Placer N '(N' = SizePop) vole à l'emplacement de recherche initial défini par le centroïde des ancres dans le rayon de transmission du capteur cible. Cette position initiale centroïde permet de se rapprocher de la cible et réduit l'espace de recherche.

$$\text{Init} (X_{\text{axis}}, Y_{\text{aix}}) = \left(\frac{1}{M} \sum_{i=1}^M x_i, \frac{1}{M} \sum_{i=1}^M y_i \right) \quad (\text{IV. 5})$$

5.3. Donner une valeur aléatoire pour la direction et de la distance de chaque individu,

$$\text{New}_i (X_{\text{axis}}, Y_{\text{aix}}) = (X_{\text{axis}} + \text{RandomValue}, Y_{\text{axis}} + \text{RandomValue}) \quad (\text{IV. 6})$$

$i = 1 \dots \text{SizePop}.$

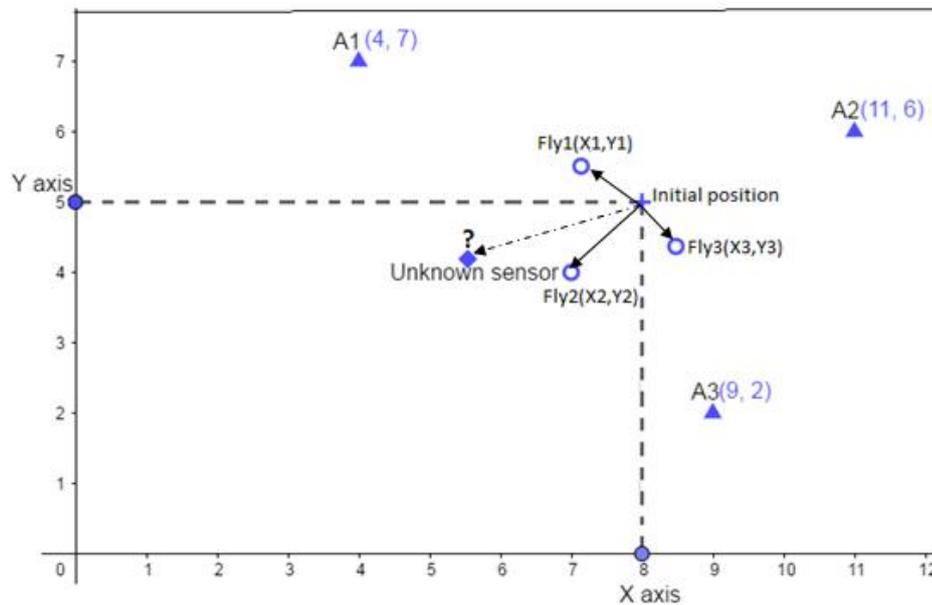


Figure IV.10 Recherche d'un nœud cible par FOA-L.

Ensuite, nous trouvons la concentration olfactive (Smell_i) de chacun des individus par la fonction objective qui représente la moyenne du carré d'erreur entre la distance estimée et la distance réelle du capteur et de l'ensemble d'ancres dans la rayon de transmission de ce capteur inconnu. La fonction objective à minimiser dans la méthode de localisation de nos capteurs est définie comme suite:

$$Smell_i = f(x, y) = \frac{1}{M} * \sum_{j=1}^M \left(\sqrt{(x - x_j)^2 + (y - y_j)^2} - \hat{d}_j \right)^2 \quad (IV. 7)$$

Où,

- $M \geq 3$ est le nombre des ancrs à un seul saut avec le nœud cible.
- (x_j, y_j) sont les coordonnées du ième ancre.
- (x, y) sont les coordonnées de la position à évaluer.
- i varie de 1 à SizePop (c'est-à-dire la taille de la population).

5.4. Nous devons trouver la mouche des fruits avec la valeur d'odeur la plus faible en utilisant la fonction objective et nous gardons sa position; ensuite, l'essaim vole vers cette position.

[bestsmell, bestindex]= Min (smell_i).

Où,

- i varie de 1 à SizePop.
- bestSmell est un optimum local représentant la meilleure solution parmi celles proposées par les mouches des fruits considérées dans cette itération.
- bestIndex est l'index de cette solution.

5.5. Répétez le processus en comparant l'odeur de la position précédente à la position actuelle, puis en conservant la meilleure jusqu'à ce que la condition d'arrêt (MaxGen) soit vérifiée.

Smellbest = bestSmell

X_{axis} = X (bestIndex)

Y_{axis} = Y (bestIndex)

5.6. Considérons la dernière position obtenue comme les coordonnées (x, y) relatives au nœud inconnu. De plus, ce capteur devient une ancre estimée. Ainsi, il diffuse un message balise contenant ses coordonnées avec un ID de nœud et des informations de signal permettant la mesure du RSSI.

5.7. Répétez les étapes 5.3 à 5.6 pour tous les nœuds localisables (N-M).

6. Après la fin du processus de localisation en utilisant FOA-L, l'erreur de localisation correspondante est calculée comme la différence entre la position estimée et la position réelle des capteurs selon l'équation suivante:

$$Error_i = \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2} \quad (IV. 8)$$

Où, (\hat{x}_i, \hat{y}_i) sont les coordonnées estimées du capteur i en utilisant la méthode FOA-L proposée, (x_i, y_i) ses coordonnées réelles et $i= 1,2, \dots, N''$ où N'' est le nombre de nœuds inconnus localisés par FOA-L. Puisque notre approche utilise les ancrs estimées dans le processus de localisation, notre méthode FOA-L permet de rendre quasiment nulle la probabilité d'avoir des nœuds capteurs non localisés à la fin des itérations.

Un critère de performance important du problème de localisation est l'erreur moyenne. Dans cette section, il est calculé par la formule suivante:

$$Error_{ave} = \frac{1}{N''} * \sum_{i=1}^{N''} Error_i \quad (IV.9)$$

À mesure que les itérations de notre méthode de localisation proposée progressent, le nombre de capteurs inconnus diminue. Par conséquent, le nombre d'ancres estimées augmente. Ces nouvelles ancrs estimées pourraient être considérées comme de nouvelles ancrs pour estimer les positions des capteurs non localisés. En effet, un capteur, qui n'est pas localisé car il n'a pas trois ancrs ou plus avec une visibilité directe dans une itération k , peut avoir plus d'ancres (c'est-à-dire de nouvelles ancrs estimées) dans l'itération $k+1$.

Nous spécifions sur la Figure (IV.11) la machine à états finis (FSM) qui illustre le processus FOA-L au sein d'un capteur avec une position inconnue. Dans l'état initial S_0 , un capteur inconnu reçoit les messages balises envoyés par les ancrs ou les ancrs estimées qui sont dans son rayon de communication. Ces messages fournissent à ce capteur l'ID de l'ancre ainsi que des informations de signal concernant ces ancrs. En fonction du nombre de messages balises reçus, le capteur inconnu passe à l'état S_1 si le nombre de messages reçus de différentes ancrs (estimées ou non) est égale à 1 ou 2. Dans l'état S_1 , le capteur est considéré comme capteur non localisable et attend la prochaine itération de FOA-L. En effet, s'il reçoit de nouveaux messages balises de différentes ancrs (c'est-à-dire des ancrs estimées) lui permettant d'atteindre au moins trois messages différents, ce capteur inconnu passe à l'état S_2 car il est considéré comme un capteur localisable. Par ailleurs, il reste dans l'état S_1 en attendant la prochaine itération du processus de FOA-L. L'état S_2 peut être atteint directement à partir de l'état initial S_0 si le capteur inconnu reçoit au moins trois messages de balise différents. À l'état S_2 , un capteur localisable utilise les valeurs RSSI calculées grâce aux messages de balise reçus pour évaluer la distance entre lui et chaque ancre correspondante. Ensuite, il exécute les étapes de FOA-L proposée afin de se localiser et passe à l'état S_3 où il devient une ancre estimée. Dans l'état S_3 , il reste dans le même état lors de l'envoi de messages balises afin de permettre aux autres capteurs de l'état S_0 ou S_1 d'avoir le nombre de messages restant permettant leur localisation.

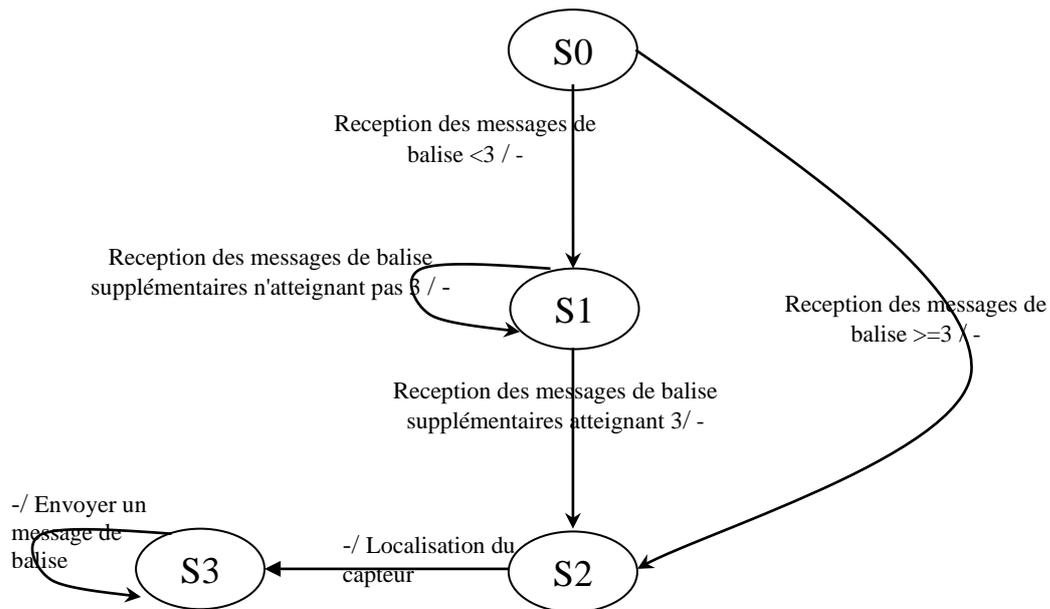


Figure IV.11 Machine à états finis des capteurs inconnus pour l’algorithme FOA-L.

Les différents messages échangés dans l'environnement de RCSF entre tous les capteurs (c'est-à-dire les ancres et les capteurs inconnus) lors de l'exécution de la méthode FOA-L sont illustrés sur le diagramme de séquence de la Figure (IV.12).

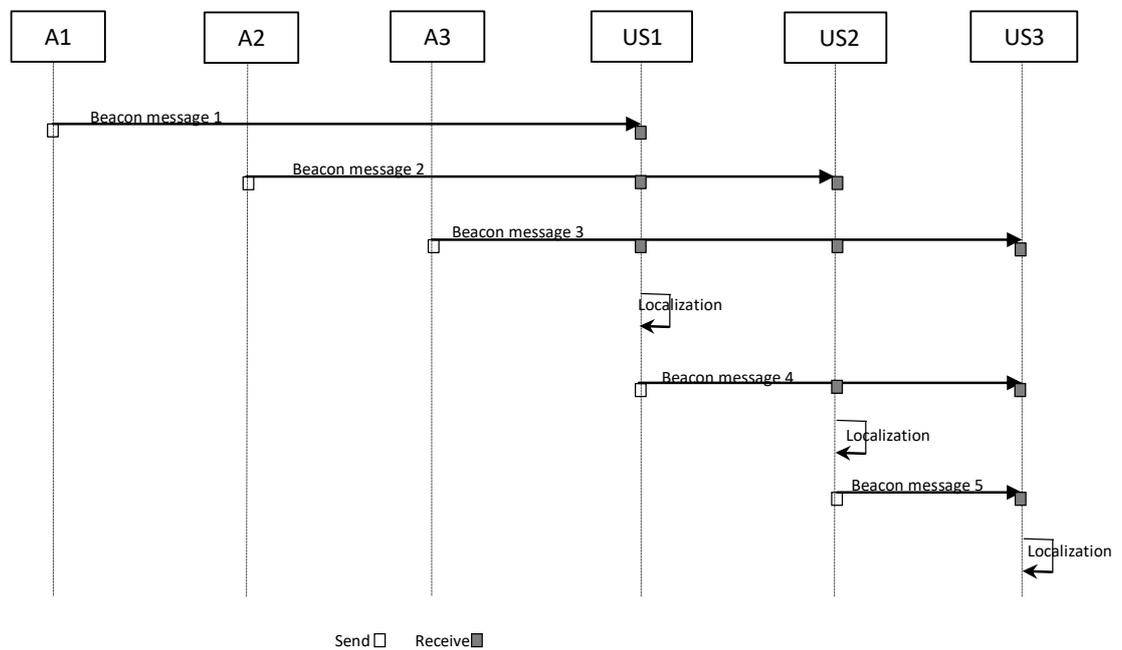


Figure IV.12 Diagramme de séquence des messages échangés entre les capteurs pour FOA-L.

La Figure (IV.12) représente tous les messages échangés entre un nombre total de 6 capteurs (trois ancres et trois capteurs inconnus) au cours d'un processus de localisation

complet. La première itération du processus FOA-L nous permet de localiser le capteur US1 (Unknown Sensor 1), après avoir reçu trois messages de balise. Cependant, le deuxième et le troisième capteur (US2 et US3) ne sont pas localisés, car ils ne disposent pas de suffisamment d'informations pour être considérés comme des capteurs localisables selon le procédé de FOA-L (deux messages balise pour US2 et un seul message pour le capteur US3).

Dans la deuxième itération, le capteur US1 devient une ancre estimée après que son emplacement a été déterminé dans l'itération précédente. Par conséquent, il envoie un message de balise comme toutes les ancres du réseau. Ainsi, nous pouvons localiser dans la deuxième itération le capteur US2 grâce à ce nouveau message reçu de l'ancre estimée, tandis que le capteur US3 attend la prochaine itération pour compléter le nombre minimum de messages balises qui lui permettent de se localiser.

Notez que nous n'utilisons pas les ancres estimées pour redéfinir les positions des nœuds inconnus, que nous avons localisés précédemment, pour éviter le coût de relocalisation. De plus, l'expérience a prouvé que cela augmente le taux d'erreur commis lors de la mise à jour de ces coordonnées. De plus, même s'il y a une petite amélioration dans de très rares cas, elle n'est pas significative par rapport à son coût élevé en termes de consommation d'énergie par exemple. Par conséquent, dans la méthode de localisation proposée FOA-L, nous utilisons les nouvelles ancres (ancres estimées) uniquement pour localiser des capteurs qui n'ont pas trois ancres ou plus comme voisins d'un seul saut.

IV.3.3 Résultats de simulation de FOA-L

Dans cette section, nous présentons différentes simulations que nous avons réalisées pour évaluer et démontrer l'efficacité de l'algorithme FOA-L. Ensuite, nous comparons les performances de cet algorithme avec des algorithmes de localisation existants et bien connus en termes d'erreur de localisation moyenne (équation IV.9) et de temps de calcul. Nous proposons plusieurs scénarios de simulation dans MATLAB pour évaluer cette proposition. La zone simulée est une topologie de grille régulière carrée de 50m×50m et le rayon de transmission des capteurs (R) est fixé à 20 mètres. Les paramètres de l'algorithme proposé sont les suivants: le nombre de population qui est égal à 20 mouches des fruits et la taille maximale des générations est égale à 100 itérations.

IV.3.3.1 Evaluation de la précision de localisation dans FOA-L

Dans le cadre de notre topologie simulée, nous évaluons l'erreur de localisation calculée par l'équation (IV.8) pour chaque nœud inconnu en utilisant notre algorithme proposé FOA-L. Les résultats obtenus de la Figure (IV.13) montrent que l'approche proposée permet d'estimer un emplacement pour tous les nœuds cibles avec une précision qui diffère d'un nœud à l'autre. Par exemple, nous avons une précision très proche des emplacements réels des nœuds 1, 2, 24 et 38 tout en ayant une plus grande erreur dans la localisation du nœud 13. Cependant, la plupart des nœuds ne dépassent pas l'erreur de localisation moyenne ($Error_{ave} = 0,09$ m) calculée par l'équation (IV.9) et qui représente la précision de notre méthode de localisation FOA-L proposée pour la configuration et la topologie considérées.

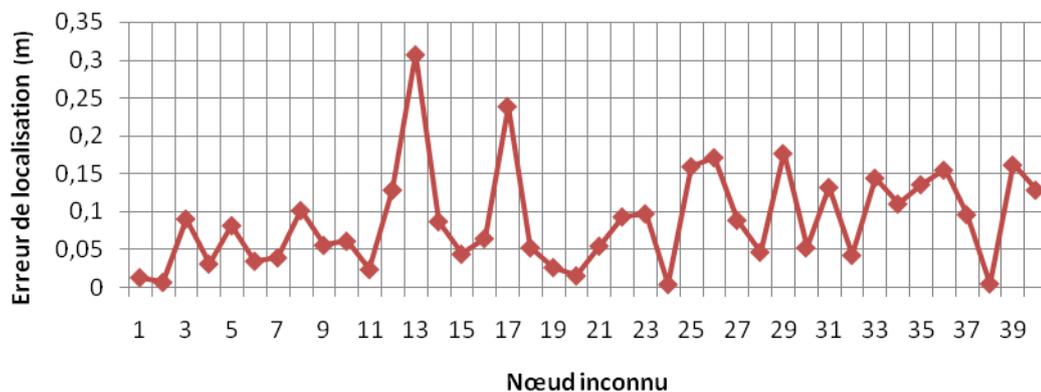


Figure IV.13 Erreur de localisation pour chaque nœud inconnu.

Afin d'évaluer l'efficacité de notre approche FOA-L tout en modifiant les paramètres de l'environnement de simulation, nous considérons des scénarios de simulation supplémentaires. Ainsi, nous ne faisons varier dans chaque scénario proposé qu'un seul paramètre afin de montrer son effet sur la précision de la localisation obtenue en utilisant notre proposition. Tout d'abord, nous fixons tous les paramètres de l'algorithme FOA-L et nous faisons varier le nombre de nœuds capteurs déployés dans la topologie considérée de 10 à 400. Les résultats obtenus sont présentés sur la Figure (IV.14). Deuxièmement, nous déployons au hasard 50 capteurs dans la topologie considérée et nous faisons varier le nombre des ancres entre 5 et 20 nœuds et nous gardons fixe tous les paramètres de la méthode de localisation. La Figure (IV.15) illustre les résultats obtenus.

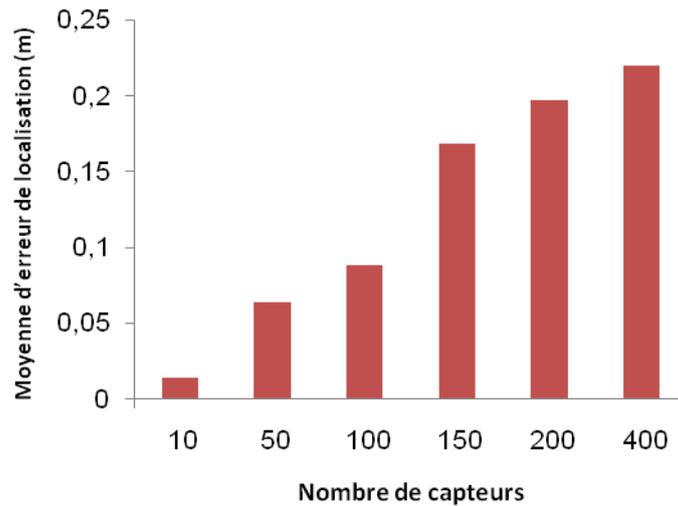


Figure IV.14 Moyenne d'erreur de localisation par rapport au nombre de capteurs.

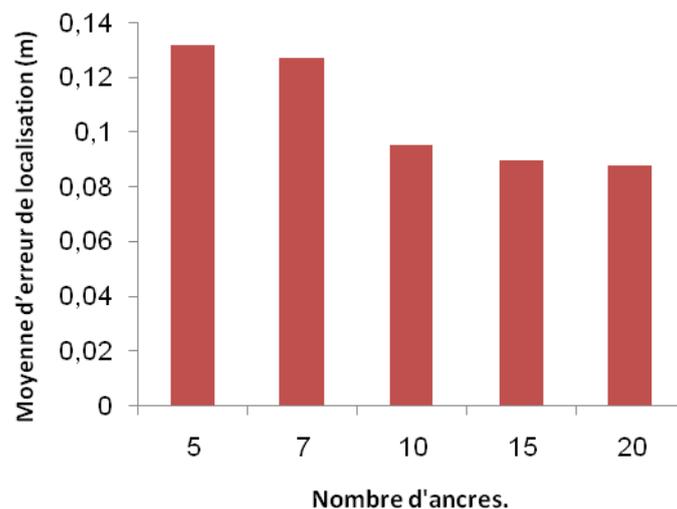


Figure IV.15 Moyenne d'erreur de localisation par rapport au nombre d'ancres.

Les résultats de simulation obtenus sur la Figure (IV.14) montrent que l'erreur moyenne de localisation obtenue par l'algorithme FOA-L proposé augmente lorsque le nombre de capteurs augmente. En effet, pour déterminer l'emplacement d'un capteur, il faut au moins la distance entre ce dernier et trois ancres (estimées ou non). Lorsque le nombre de capteurs augmente dans notre topologie tout en conservant le même nombre des ancres, les ancres estimées (avec les positions estimées) utilisées pour localiser les nœuds qui n'ont pas trois ancres ou plus dans leur visibilité direct augmenteront l'erreur de localisation moyenne comme le montre la Figure (IV.14).

D'autre part, les résultats de simulation obtenus sur la Figure (IV.15) montrent que l'erreur moyenne de localisation de la méthode proposée diminue lorsque le nombre des

ancres augmente. Ensuite, il se stabilise lorsque le nombre d'ancres atteint 10 et diminue légèrement jusqu'à 20 ancres. En effet, un capteur inconnu doit recevoir trois messages de balise de trois ancres différentes afin de se localiser, ce qui est rendu possible à partir d'un nombre spécifique d'ancres au sein d'une topologie fixe.

Enfin, nous évaluons l'effet du changement du nombre de mouches tout en utilisant cette méthode de localisation proposée. On utilise donc les paramètres ayant donné les meilleurs résultats dans l'étude précédente et on fait varier le nombre de mouches de 5 à 40. Les résultats obtenus de la Figure (IV.16) montrent l'effet d'augmenter le nombre de mouches sur la moyenne de l'erreur de localisation lors de l'utilisation de la méthode FOA-L. On remarque que l'utilisation de 20 mouches garantit presque un bon résultat. De plus, en obtenant le nombre suffisant de mouches proportionnel à la configuration du réseau, nous obtiendrons presque la même précision de localisation.

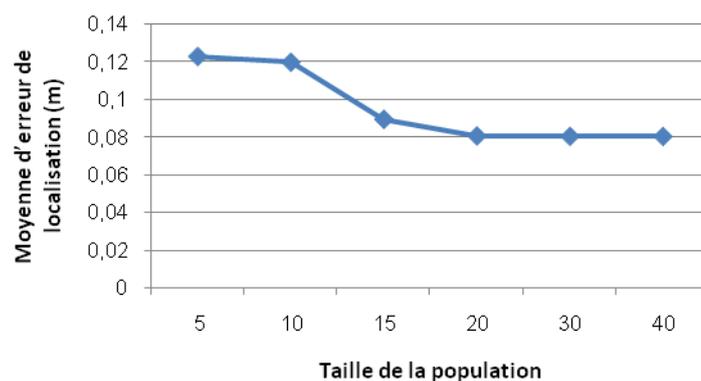


Figure IV.16 Moyenne d'erreur de localisation par rapport à la taille de la population.

IV.3.3.2 Comparaison de FOA-L avec des travaux similaires

Afin de démontrer l'efficacité de notre proposition FOA-L, nous la comparons à d'autres travaux de recherche connexes permettant la localisation par des méta-heuristiques au sein des réseaux de capteurs sans fil. Afin d'assurer une comparaison équitable, nous utilisons les mêmes paramètres de configuration que ceux présentés dans le Tableau (IV.2). Dans notre étude comparative, nous évaluons les performances de l'algorithme FOA-L ainsi que deux algorithmes bio-inspirés: PSO (Particle Swarm Optimization) et CSO (Chicken Swarm Optimization) présentés dans [Al Shayokh et Soo, 2017]. Cette étude comparative concerne non seulement l'erreur de localisation moyenne mais également le temps d'exécution lors de l'utilisation de ces approches de localisation. Les résultats

obtenus (voir le Tableau (IV.3)) montrent que notre algorithme FOA-L proposé fonctionne mieux que les deux autres approches. En effet, FOA-L permet, avec un temps de calcul faible, une précision de localisation meilleure que l'algorithme CSO ainsi que l'algorithme PSO, qui a prouvé son efficacité pour la localisation dans de nombreux travaux de recherche antérieurs comme dans [Al Shayokh et Soo, 2017]. Les caractéristiques particulières par rapport à PSO et CSO, et qu'ils ont un modèle de mouvement d'essaim plus compliqué. De plus, dans la méthode FOA-L proposée, tous les individus de l'essaim se sont réunis à chaque fois dans la position du meilleur individu, permettant une recherche approfondie de l'emplacement possible du capteur. En se basant sur le centroïde comme position initiale, FOA-L permet d'économiser un temps de recherche important, par rapport à CSO, par exemple, qui utilise des emplacements aléatoires comme position initiale pour la recherche, et peut donc rester longtemps avant de se rapprocher de la bonne solution.

Tableau IV.2: Paramètres de configuration du scénario de simulation.

Paramètres	Valeur
Zone	100m×100m
Rayon de transmission	25m
Nombre de capteurs	50 capteurs
Nombre des ancrs	10 capteurs
Taille de la population	20 individu
Génération maximale	150 itérations

Tableau IV.3: Résultats de la comparaison.

Algorithme	Moyenne de l'erreur de localisation (m)	Temps d'exécution (s)
PSO	0.375	874.24
CSO	0.110	255.65
FOA-L	0.092	121.10

IV.3.4 Récapitulatif

Comme tous les problèmes d'ingénierie et d'intelligence artificielle, le problème de la localisation dans les réseaux de capteurs sans fil peut être modélisé comme un problème d'optimisation global. Récemment, les chercheurs ont largement utilisé les méta-heuristiques pour résoudre le problème de l'optimisation de la localisation dans les RCSF. Dans cette proposition, l'algorithme d'optimisation de Fruit Fly est appliqué pour réduire

l'erreur de localisation entre les coordonnées réelles et estimées des capteurs avec des positions inconnues. Les résultats de la simulation montrent que notre algorithme proposé (FOA-L) offre une meilleure précision de localisation dans un temps de calcul raisonnable en comparaison avec des algorithmes similaires tels que PSO et CSO. D'autre part, nous avons démontré que la précision de localisation obtenue lors de l'utilisation de la méthode FOA-L est influencée par les paramètres et les configurations utilisés du réseau de capteurs sans fil considéré.

IV.4 Conclusion

DV-Hop(FOA), DV-Distance(FOA) et FOA-L sont de nouvelles approches pour la localisation des nœuds dans les réseaux de capteurs sans fil. Elles sont basées sur l'algorithme FOA inspiré du comportement de recherche de nourriture observé chez les Fruits Fly réelles. L'intérêt principal de ces algorithmes est leurs capacités à déterminer automatiquement les positions de différents capteurs inconnues dans un RCSF en utilisant l'information sur les positions de peu de capteurs appelés des ancres avec la meilleure précision possible. Dans DV-Hop(FOA) et DV-Distance(FOA) l'algorithme d'optimisation FOA est adapté pour optimiser la précision de deux techniques de localisation existantes (DV-HOP et DV-DISTANCE), l'hybridation était de permettre d'atteindre cet objectif. Dans FOA-L, on a pu savoir que FOA est efficace à localiser les différents capteurs inconnus.

Les expériences effectuées montrent que ces algorithmes, DV-Hop(FOA), DV-Distance(FOA) et FOA-L, fournissent de très bons résultats et s'avèrent comme prévu supérieurs aux algorithmes existants.

Conclusion générale

Les réseaux de capteurs sans fil sont devenus une technologie clé et sont utilisés dans de plus en plus de problèmes industriels et environnementaux. Cependant, avant de voir apparaître un réseau de capteurs comme un monde « tous capteurs », un certain nombre de problématiques doit être résolu. La définition d'un algorithme de localisation efficace est le premier défi pour la mise en place des réseaux de capteurs déployés aléatoirement. L'étude de la littérature a montré qu'un algorithme de localisation optimal ne pouvait pas encore être défini, et donc le choix de l'algorithme de localisation doit être fondé principalement sur les résultats escomptés.

Le thème de notre thèse est l'optimisation de la localisation en utilisant les méta-heuristiques bio-inspirées. L'objectif visé est à la fois, l'adaptation des méta-heuristiques pour l'optimisation des techniques de localisation existantes ainsi que la proposition de nouveaux algorithmes de localisation.

Les méthodes méta-heuristiques peuvent apporter un plus à des problèmes d'optimisation d'une part et d'améliorer les performances de la localisation d'autre part. Les techniques traditionnelles utilisées pour la localisation dans les RCSFs, ont des limitations dans la construction d'approches capables d'arriver à des solutions suffisantes en termes de précision ainsi que le temps d'exécution qui influence directement les ressources.

Toutefois, notre étude a pour but de présenter une méthode méta-heuristique comme une méthode de résolution approchée simple et adaptable à tout type de problèmes de localisation dans les RCSFs. La capacité de ces méta-heuristiques est d'optimiser le problème de localisation avec un minimum d'informations sur les positions de quelques capteurs appelés des ancres, et elle est équilibrée par le fait qu'elle n'offre aucune garantie quant à l'optimalité totale de la solution trouvée. Du point de vue de la recherche opérationnelle et de l'optimisation combinatoire, cet inconvénient n'est pas toujours un problème, tout spécialement quand une seule approximation de la solution optimale est recherchée.

Nos approches de localisation sont basées sur l'algorithme d'optimisation par Fruit Fly (Fruit Fly Optimization Algorithm, i.e. FOA). Poursuivant plus loin avec l'une des idées proposées, le FOA nous a permis d'affiner les positions initiales des nœuds inconnus estimées par une célèbre techniques de localisation DV-Hop. Les résultats prometteurs trouvés par cette hybridation, nous ont encouragés de faire le même travail avec DV-distance. Cela a conduit à deux développements majeurs qui ont été proposés dans la thèse (DV-Hop(FOA) et DV-distance(FOA)), et dont les améliorations augmentent considérablement la précision de positionnement. Ils ne nécessitent pas beaucoup plus de coûts de calcul et correspondent parfaitement aux exigences de l'algorithme distribué.

Un troisième aspect des travaux concernait la faisabilité d'approche basée seulement sur la méta-heuristique FOA. La simulation dans la Plateforme Matlab et les comparaisons avec d'autres méta-heuristiques de localisation performantes nous a fourni de précieuse vision concernant la performance de notre proposition (FOA-L) pour la localisation. Bien que la carte de la localisation ait donné des informations prometteuses sur les possibilités de positionnement par des méthodes bio-inspirées, les expériences ont montré que les valeurs sont vraiment sensibles à l'environnement et à la nature des perturbations, qui sont encore plus importantes dans une configuration intérieure.

En raison de la quantité importante de scénarios où les processus de localisation entrent en jeu, il a été et continuera d'être un domaine de recherche important. Les différents types d'algorithmes pourraient être améliorés et de nouveaux développés, mais un autre problème sera de déterminer quel algorithme pourrait donner les résultats les plus intéressants, pour une configuration donnée. En tant que travaux futurs concernant nos propositions, nous visons à étendre l'utilisation dans le contexte de l'architecture de réseau de capteurs mobiles ainsi que dans la topologie 3D.

Contribution : Publication et travaux réalisés

Avant la phase doctorale

- Rabhi Seddik, Fizazi hadria et Benyamina Ahmed. conception d'une approche biomimétique hybride firefly et k-means pour la classification des images, *international conference on electrical engineering Bechar Alegria*, 17 et 19 Novembre 2013
- Rabhi Seddik, Fizazi hadria et Benyamina Ahmed. A hybrid approach biomimetic for images clustering inspiration of the fireflies natureal 2nd international conference on signal, image, vision and their applications Guelma algeria (SIVA'2013) November 18-20 2013
- Rabhi Seddik, Fizazi hadria. Etude comparative entre l'algorithme de Firefly et les comportements collectifs des fourmis dans la classification des Images satellitaires 3ème international conférence on vision artificiel Tizi-Ouzou algeria (CVA'2015) Avril 12-14 2015
- Rabhi Seddik, Fizazi hadria. Regroupement des données de télédétection en utilisant les techniques bio-inspirées 2ème Conférence internationale sur le traitement de l'information multimédia (CITIM'2015) Mai 12-13 2015

Durant la phase doctorale

Publication dans une revue internationale

Rabhi, S., Semcheddine, F. (2019). Localization in wireless sensor networks using DV-hop algorithm and fruit fly meta-heuristic. *Advances in Modelling and Analysis B*, Vol. 62, No. 1, pp. 18-23. https://doi.org/10.18280/ama_b.62010

Article accepté pas encore en ligne

Rabhi, S. Semcheddine, F. Mbarek, N. An Improved Method for Distributed Localization in WSNs based on Fruit Fly Optimization Algorithm. *Aut. Control Comp. Sci.* xx, xx-xx (202x).

Journal of Automatic Control and Computer Sciences

Articles soumis

- ✓ En révision dans une conférence internationale sous le titre :
Hybrid Localization Approach Based on DV-Distance and Fruit Fly Optimization Algorithm for WSNs.

Encadrements : Mémoire Master LMD soutenu (lié au thème de la thèse)

- BELBALI Mohammed Said & HAISSOUNE Mahfoud « Optimisation d'une technique de localisation des nœuds dans les réseaux de capteurs sans fil » Université d'Adrar 2018-2019
- BADDOU Salih & CHETOIAH Karima « Application de l'essaïn de poulet pour la localisation dans les RCSF » Université d'Adrar 2017-2018

Références bibliographiques

[Akyildiz et al, 2002] Ian F. Akyildiz ; W.Su ; Y.Sankarasubramaniam ; and E. Cayirci. A survey on sensor networks. volume 4, pages 393–422. IEEE Communications Magazine, Août 2002.

[Al Shayokh et Soo, 2017] Al Shayokh, Md, and Soo Young Shin. "Bio Inspired Distributed WSN Localization Based on Chicken Swarm Optimization." *Wireless Personal Communications* 97.4: 5691-5706.2017).

[Allaoua et Laoufi, 2009] Allaoua B. & Laoufi A., « Optimal Power Flow Solution Using Ant Manners for Electrical Network », *Advances in Electrical and Computer Engineering*, Vol. 9, N°. 1,2009.

[Amundson et Koutsoukos, 2009] I.Amundson ; X. D. Koutsoukos. A survey on localization for mobile wireless sensor networks. *International Workshop on Mobile Entity Localization (MELT)*, septembre 2009.

[Bahl et Padmanabhan, 2000] Bahl, P., & Padmanabhan, V. N. (2000). RADAR: An in-building RF-based user location and tracking system. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (Vol. 2, pp. 775-784). DOI: 10.1109/INFCOM.2000.832252.*

[Berthiau et Siarry, 2001] Berthiau, Gerard & Siarry, Patrick. (2001). État de l'art des méthodes "d'optimisation globale". *RAIRO - Operations Research*. 35. 329 - 365. 10.1051/ro:2001118.

[Biswas et Ye, 2004] Biswas, P., & Ye, Y. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks* (pp. 46-54). ACM. (2004, April).

[Bontempi, 2006] G. Bontempi. *Statistical Foundations of Machine Learning*. Université Libre de Bruxelles, 2006.

[Boukerche, 2008] Azzedine Boukerche, —*Algorithms And Protocols For Wireless Sensor Networks*], Wiley, ISBN: 978-0-471-79813-2, 11-2008.

[Boushaba et al, 2009] Boushaba, M., Hafid, A., & Benslimane, A. (2009). High accuracy localization method using AoA in sensor networks. *Computer Networks*, 53(18), 3076-3088. DOI:10.1016/j.comnet.2009.07.015.

[Brown et al, 2007] Brown, C., Liebovitch, L. S., & Glendon, R. (2007). Lévy flights in DobeJu/'hoansi foraging patterns. *Human Ecology*, 35, 129–138.

[Bulusu et al, 2000] Bulusu, N., Heidemann, J., Estrin, D.:GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, vol. 7, no. 5, Oct. 2000, pp. 28-34

[Cao et al, 2012] Cao, S., Wang, J., & Gu, X. A wireless sensor network location algorithm based on firefly algorithm. In *AsiaSim 2012* (pp. 18-26). Springer, Berlin, Heidelberg. (2012).

- [**Cardei et al, 2005**] Mihaela Cardei, Ionut Cardei, Ding-Zhu Du, —Resource Management in Wireless Networking, Springer, ISBN: 978-0-387-23807-4, 01-2005
- [**Černý , 1985**] V. Černý. A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41-51, 1985
- [**CHALLAL, 2008**] CHALLAL. Yacine. Réseaux de Capteurs Sans Fils : Support de cours. Ecole nationale Supérieure d'Informatique, Algérie. Novembre 2008. <https://moodle.utc.fr/file.php/498/support-SIT60.pdf>
- [**Chen et al, 2008**] H.Chen, K.Sezaki, P.Deng, and H.C.So, "An improved DV-hop localization algorithm for wireless sensor networks," in *Proceedings of IEEE Conference on Industrial Electronics and Applications*, Singapore, 2008, pp.1557-1561.
- [**Cho et al, 2011**] Cho, H. H., Lee, R. H., & Park, J. G. (2011). Adaptive parameter estimation method for wireless localization using RSSI measurements. *Journal of Electrical Engineering and Technology*, 6(6), 883-887. DOI:10.1007/978-3-642-22309-9_29
- [**Cuevas et al, 2010**] Cuevas.E, Zaldivar.D ,Valentine.M. A novel multi-threshold segmentation approach based on differential evolution optimization. *Expert systems with application* Vol.37, pp.5265-5271.2010.
- [**Dargie et Poellabauer, 2010**] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*, John Wiley & Sons, 2010.
- [**Deng et al, 2008**] B.Deng H.Liu L.Zhang, L.Duan and G.Huang. S-mrl distributed localization algorithm for wireless sensor networks. *Wireless Communications, Networking and Mobile Computing*, 2008. WiCOM '08. 4th International Conference on, pages 1–4, 2008.
- [**Dhurandher et al, 2008**] S. Dhurandher, S. Misra, M. S. Obaidat and S. Khairwal, "UWSim: a simulator for underwater wireless sensor networks," *Simulation: Transactions of the Society for Modeling and Simulation International, SCS*, Vol. 84, No. 7, pp. 327–338, July 2008.
- [**Ding et al, 2012**] Ding, J., Zhang, L., Cheng, G., Ling, Z., Zhang, Z., & Lei, Y. (2012). Study on DV-Hop algorithm based on modifying hop count for wireless sensor networks. *International Journal of Computer Science Engineering & Technology* . Vol. 2 Issue 10, p1452-1456. 5p.
- [**Dorigo et Caro, 1999**] Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on (Vol. 2, pp. 1470-1477)*. IEEE. DOI: 10.1109/CEC.1999.782657.
- [**Dorigo et Stutzle, 2004**] M. Dorigo And T. Stützle. *Ant Colony Optimization*, Cambridge: MIT Press, 2004.
- [**Dorigo, 1992**] Dorigo, M.: *Optimization, learning and natural Algorithms*. Ph.D. thesis, Politecnico di Milano, Italy (1992)
- [**Dorigo, 2007**] M. Dorigo. *Ant Colony Optimization*, Scholarpedia, Vol.2, N°03,ppn 1461.http://www.scholarpedia.org/article/Ant_colony_optimization, 2007.

- [**Dreo and Siarry, 2003**] Dreo, J. and P. Siarry. Diverses techniques d'optimisation inspirées de la théorie de l'auto-organisation dans les systèmes biologiques. In . Paris, France. 2003.
- [**Farmer et al, 1992**] Farmer, J.D., Packard, N.H., Perelson, A.S.: The immune system, adaptation, and machine learning. *Phys. D* 2(1–3), 187–204 (1986)
- [**Feo et Resende, 1989**] Feo, T.A., Resende, M.G.C.: A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters* 8(2), 67–71 (1989)
- [**Feyel, 2015**] Philippe Feyel. Optimisation des correcteurs par les métaheuristiques. Application à la stabilisation inertielle de ligne de visée. Autre. CentraleSupélec, 2015. Français. NNT : 2015CSUP0011. tel-01260457v2.
- [**Fonseca et Fleming, 1993**] Fonseca, C. M., & Fleming, P. J. (1993, June). Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization. In *Icga* (Vol. 93, No. July, pp. 416-423).
- [**Geem et al, 2001**] Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: Harmony search. *Simulation* 76(2), 60–68 (2001)
- [**Girod et al, 2002**] Girod, L., Bychkovskiy, V., Elson, J., & Estrin, D. (2002). Locating tiny sensors in time and space: A case study. In *Computer Design: VLSI in Computers and Processors, 2002. Proceedings. 2002 IEEE International Conference on* (pp. 214-219). IEEE. .DOI: 10.1109/ICCD.2002.1106773.
- [**Glover, 1986**] Glover, F.: Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* 13(5), 533–549 (1986)
- [**Goldberg, 1989**] Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1st edn. Studies in Computational Intelligence. Addison-Wesley Longman (1989)
- [**Gopakumar et Jacob, 2008**] Gopakumar, A., & Jacob, L. Localization in wireless sensor networks using particle swarm optimization. (2008).
- [**Goyal et Patterh, 2014**] Goyal, S., & Patterh, M. S. Wireless sensor network localization based on cuckoo search algorithm. *Wireless personal communications*, 79(1), 223-234. (2014).
- [**Gui et al, 2015**] L.Gui , T. Val, A. Wei, and R. Dalce,” Improvement of range-free localization technology by a novel DV-hop protocol in wireless sensor networks”, *Ad Hoc Networks*, vol. 24, pp.55-73, 2015.
- [**Hall, 1999**] Prentice Hall, T.S. Rappaport. *Wireless Communications, Principles & Practices*1999.
- [**Hansen et al, 1995**] Hansen, N., Ostermeier, A., Gawelczyk, A.: On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In: *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 57–64. Morgan Kaufmann, San Francisco (1995).

[Harter et al, 2002] A. Harter, A. Hopper, P. Steggles, A. Ward and P. Webster, 2002. The anatomy of a context-aware application. *Wireless Networks*, 8: pp. 187-197. <https://doi.org/10.1023/A:1013767926256>.

[He et al, 2003] He, T., Huang, C., Blum, B. M., Stankovic, J. A., & Abdelzaher, T. (2003, September). Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*(pp. 81-95). ACM. Doi:10.1145/938985.938995.

[He et al, 2006] He, S., Wu, Q., Saunders, J.: A novel group search optimizer inspired by animal behavioural ecology. In: *Proceedings of 2006 IEEE Congress on Evolutionary Computation*, pp. 16–21, Vancouver (2006).

[Hightower et, Borriello 2001] Jeffrey Hightower and Gaetano Borriello, *Location Systems for Ubiquitous Computing*, IEEE Computer Journal, VOL. 34 N° 8, pp. 57-66, 2001.

[Hofmeyr et Forrest, 2000] Hofmeyr, S. A., & Forrest, S. (2000). Architecture for an artificial immune system. *Evolutionary computation*, 8(4), 443-473. DOI:10.1162/106365600568257.

[Holland, 1975] Holland. J. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor. 1975.

[Huang et al, 2003] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. ACM Press New York, NY, USA, 2003.

[Ji et Zha, 2004] Ji, X., & Zha, H. (2004, March). Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies* (Vol. 4, pp. 2652-2661). IEEE. DOI: 10.1109/INFCOM.2004.1354684.

[Kannan et al, 2006] Kannan, A. A., Mao, G., & Vucetic, B. Simulated annealing based wireless sensor network localization. *JCP*, 1(2), 15-22. (2006).

[KARA, 2009] Messaoud KARA, Réseau de capteurs sans fil : Etude en vue de la réalisation d'un récepteur GPS différentiel à faible coût. THÈSE de doctorat; Université Blaise Pascal – Clermont-Ferrand II. 2009.

[KARABOGA et BASTURG, 2007] KARABOGA D., BASTURG B., « A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm », *Journal of Global Optimization*, Springer, Vol. 39, Issue 3, p. 459-471, Avril 2007.

[Karl et Willig, 2005] Holger Karl, Andreas Willig, —*Protocols And Architectures For Wireless Sensor Networks*], Wiley, ISBN: 978-0-470-09510-2, 04-2005

[Kaur et al, 2017] Kaur, Amanpreet & Gupta, Govind & Kumar, Padam. (2017). A Survey of Recent Developments in DV-Hop Localization Techniques for Wireless Sensor Network. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*. 9. 61-71.

- [**Kennedy et Eberhart, 1995**] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995).
- [**Kirkpatrick et al, 1983**] Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. *Science* 220(4598), 671–680 (1983)
- [**Kirkpatrick et al, 1983**] S.Kirkpatrick, C. D. Gelatt, and M. P.Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [**Koza, 1992**] Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems), 1st edn. MIT Press (1992).
- [**Kulaib et al, 2011**] A. R. Kulaib, R. M. Shubair, M. A. Al-Qutayri, and J. W. P. Ng, ‘An overview of localization techniques for wireless sensor networks,’ in Proc. Int. Conf. Innov. Inf. Technol. (IIT), Apr. 25–27, 2011, pp. 167–172.
- [**Laptik et Navakauskas, 2007**] Laptik.R, Navakauskas.D. Application of Ant Colony Optimization for Image Segmentation. *Electronics and Electrical Engineering*. Kaunas: Technologija, Vol.80. No. 8. pp. 13–18.2007.
- [**Li et Kunz, 2007**] Li, L., & Kunz, T. (2007, October). Localization applying an efficient neural network mapping. In Proceedings of the 1st international conference on Autonomic computing and communication systems (p. 1). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). <http://dx.doi.org/10.4108/ICST.AUTONOMICS2007.2126>.
- [**Lin et al, 2005**] K. Lin, J. Yu, J. Hsu, S. Zahedi, D. Lee, J. Friedman, A. Kansal, V. Raghunathan, and M. Srivastava. ‘Heliomote: enabling long-lived sensor networks through solar energy harvesting’. In SenSys ‘05: Proceedings of the 3rd international conference on Embedded networked sensor systems, pages 309_309, New York, NY, USA, 2005.ACM.
- [**MacArthur et Wilson, 1967**] MacArthur, R., Wilson, E.: The Theory of Biogeography. Princeton University Press, Princeton, NJ (1967).
- [**Marks et Niewiadomska, 2007**] Marks, M., & Niewiadomska-Szynkiewicz, E. Two-phase stochastic optimization to sensor network localization. In Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on (pp. 134-139). IEEE. (2007, October).
- [**Meng et al, 2014**] Meng, X., Liu, Y., Gao, X., & Zhang, H. (2014). A new bio-inspired algorithm: Chicken swarm optimization. In Advances in swarm intelligence (pp. 86–94). Springer, Berlin.
- [**Metropolis et al, 1953**] KARABOGA D., BASTURG B., « On the performance of artificial bee colony (ABC) algorithm », *Journal Applied Soft Computing*, Elsevier, Vol. 8, Issue 1, p.687-697, Janvier 2008.
- [**Metropolis et al, 1953**] Meng, X., Liu, Y., Gao, X., & Zhang, H. (2014). A new bio-inspired algorithm: Chicken swarm optimization. In Advances in swarm intelligence (pp. 86–94). Springer, Berlin.
- [**Metropolis et al, 1953**] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. "Equations of State Calculations by Fast Computing Machines". *Journal of Chemical Physics*, 21(6):1087- 1092, 1953.

[**Moussa et El-Sheimy, 2010**] A. Moussa and N. El-Sheimy, "Localization of wireless sensor network using Bees Optimization Algorithm," The 10th IEEE International Symposium on Signal Processing and Information Technology, Luxor, 2010, pp. 478-481.

[**Mühlenbein et Paaß, 1996**] Mühlenbein, H., Paaß, G.: From recombination of genes to the estimation of distributions. I.Binary parameters. In: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, PPSN IV, pp. 178–187. Springer, London (1996).

[**Nan et al, 2007**] Nan, G. F., Li, M. Q., & Li, J. Estimation of node localization with a real-coded genetic algorithm in WSNs. In Machine learning and cybernetics, 2007 International Conference on (Vol. 2, pp. 873-878). IEEE. (2007, August).

[**Niculescu et Nath, 2001**] Niculescu D, Nath B. "Ad-Hoc positioning systems (APS)". Proceedings of the 2001 IEEE Global Communications Conference (GCC'01). New York USA: IEEE, 2001, pp. 2926-2931.

[**Nicopolitidis et al, 2003**] P. Nicopolitidis, M. S. Obaidat, G. I. Papadimitriou and A. S. Pomportsis, Wireless Networks, John Wiley & Sons, 2003.

[**Niculescu et Nath, 2003**] Niculescu, D., & Nath, B. (2003). DV based positioning in ad hoc networks. Telecommunication Systems, 22(1-4), 267-280. DOI:10.1023/A:1023403323460

[**Obaidat et al, 2009**] M. S. Obaidat, P. Nicopolitidis and J.-S. Li, "Security in wireless sensor networks," Security and Communications Networks, Wiley, Vol. 2, No. 2, pp. 101–103, March/April 2009.

[**Onwubolu et Babu, 2004**] Onwubolu G. C., and Babu B. V. New Optimization Techniques in Engineering, Springer-Verlag, Germany.2004.

[**Pan, 2011**] Pan, W. T. (2011). A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowledge-Based Systems. 2011, 26. pp: 69-74. <https://doi.org/10.1016/j.knosys.2011.07.001>

[**Passino, 2002**] Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine 22(3), 52–67 (2002). doi:10.1109/MCS.2002.1004010

[**Paul et Sato, 2017**] Paul, A. K., & Sato, T. (2017). Localization in Wireless Sensor Networks: A Survey on Algorithms, Measurement Techniques, Applications and Challenges. Journal of Sensor and Actuator Networks, 6(4), 24. <https://doi.org/10.3390/jsan6040024>

[**Pavlyukevich, 2007**] Pavlyukevich, I. (2007). Cooling down Lévy flights. Journal of Physics A, Mathematical and Theoretical, 40, 12299–12313.

[**Peng et Li, 2015**] B. Peng, and L. Li, "An improved localization algorithm based on genetic algorithm in wireless sensor networks", Cognitive Neurodynamics, vol. 9, no.2, pp. 249-256, 2015.

[**Price et al., 2005**] Price V. Kenneth., Storn M. Rainer., and Lampinen A. Jouni. Differential evolution: A practical approach to global optimization. Springer-Verlag Berlin, Heidelberg.2005.

- [**Rabhi et Fizazi, 2015**] S.Rabhi, H.Fizazi. Étude comparative entre l’algorithme de firefly et les comportements collectifs des fourmis dans la classification des images satellitaire. Troisième conférence sur la vision artificielle CVA 2015. Tizi ouzou. Algérie. 2015.
- [**Raghunathan et al, 2002**] V. Raghunathan ; C. Schurgers ; S. Park ; and M. B. Srivastava. Energy-aware wireless microsensor networks. IEEE Signal Processing Magazine, Volume: 19 , Issue: 2 , Mar 2002.
- [**Rashedi et al, 2009**] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: A gravitational search algorithm. Information Sciences 179(13), 2232–2248 (2009). doi:10.1016/j.ins.2009.03.004
- [**Saad, 2008**] Clément Saad. Quelques contributions dans les réseaux de capteurs sans fil : Localisation et Routage. Réseaux et télécommunications [cs.NI]. Thèse de doctorat, Université d’Avignon, 2008. Français.
- [**Safa, 2014**] H.Safa, “A novel localization algorithm for large scale wireless sensor networks”, Computer Communications, vol. 45, pp 32-46, 2014.
- [**Sassi et al, 2014**] Sassi, H., Najeh, T., & Liouane, N. (2014). A Selective 3-Anchor DV-Hop Algorithm Based On the nearest anchor for wireless sensor network. World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 8(10), 1756-1760.
- [**Savvides et al, 2002**] A. Savvides, H. Park, et M. Srivastava, 2002. The bits and flops of the n-hop multilateration primitive for node localization problems. First ACM International WSNA, Atlanta.
- [**Shlesinger et Zaslavsky, 1995**] Shlesinger, M. F., Zaslavsky, G.M., & Frisch, U. (Eds.). (1995). Lévy flights and related topics in physics. Berlin: Springer.
- [**Shlesinger, 2006**] Shlesinger, M. F. (2006). Search research. Nature, 443, 281–282.
- [**Simic et Sastry, 2003**] S. Simic and S. Sastry. A distributed algorithm for localization in random wireless networks. Discrete Applied Mathematics, 2003.
- [**Simon, 2008**] Simon, D.: Biogeography-based optimization. IEEE Transactions on Evolutionary Computation 12(6), 702–713 (2008)
- [**Sivakumar et Venkatesan, 2016**] S. Sivakumar & Raman Venkatesan. Error Minimization in Localization of Wireless Sensor Networks using Ant Colony Optimization. International Journal of Computer Applications 145(8):15-21 (July 2016)
- [**Sohraby et al, 2007**] K. Sohraby, D. Minoli and T. Znati, Wireless Sensor Networks: Technology, Protocols, and Applications, John Wiley & Sons, 2007.
- [**Song et Tam, 2015**] G. Song, and D. Tam,” Two Novel DV-Hop Localization Algorithms for Randomly Deployed Wireless Sensor Networks”, International Journal of Distributed Sensor Networks, 2015.
- [**Srikanth et al, 2018**] Banda Srikanth, Hemant Kumar, Karanam U.M Rao (2018). A robust approach for WSN localization for underground coal mine monitoring using improved RSSI technique. Mathematical Modelling of Engineering Problems. vol. 5, No. 3, September, 2018, pp. 225-231 DOI:10.18280/mmep.050314

- [**Stojmenovic, 2005**] Ivan Stojmenovic, —Handbook Of Sensor Networks: Algorithms And Architectures, Wiley-Interscience, ISBN: 978-0-471-68472-5, 10-2005
- [**Storn et Price, 1997**] R. Storn and Price. K. Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization Vol.11, pp 341–359, 1997.
- [**Storn et Price, 1997**] Storn, R.M., Price, K.V.: Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
- [**Su et al, 2001**] Y. Sankarasubramaniam E. Cayirci I.F. Akyildiz, W. Su*. Wireless sensor networks: a survey. volume 4, pages 393–422, December 2001.
- [**Tomic et Mezei, 2016**] S. Tomic, and I. Mezei, "Improvements of DV-Hop localization algorithm for wireless sensor networks." Telecommunication Systems, pp. 1-14, 2016.
- [**Törn et Zilinskas, 1989**] A. Törn et A. Zilinskas, Global optimization, 'edit'e par G. Goos et J. Hartmanis. Springer Verlag, No. 350 (1989).
- [**Walker et al, 1993**] Walker, A., Hallam, J., Willshaw, D.: Bee-havior in a mobile robot: The construction of a self-organized cognitive map and its use in robot navigation within a complex, natural environment. In: Proceedings of ICNN'93, International Conference on Neural Networks,
- [**WANG et al, 2010**] WANG, Dianhong, JIA, Hongdong, CHEN, Fenxiong, et al. An improved DV-Distance localization algorithm for wireless sensor networks. In : 2010 2nd International Conference on Advanced Computer Control. IEEE, 2010. p. 472-476.
- [**Yang , 2008**] Yang X. S. Nature-Inspired Metaheuristic Algorithms. Luniver Press, 2008.
- [**Yang et Deb, 2009**] Yang, X., & Deb, S. (2009). Cuckoo search via Lévy flights. In World congress on nature & biologically inspired computing (NaBIC2009), IEEE, 978-1-4244-5612-3/09.
- [**Yang, 2009**] Yang X.S, Firefly algorithm for multimodal optimization in proceedings of the stochastic Algorithms, SAGA 109. Vol.5792, Oct.2009.
- [**Yang, 2010**] Yang, X. S. Firefly Algorithm, Stochastic Test Functions and Design Optimisation . Int. J. Bio-Inspired Computation, Vol. 2, No. 2, pp.78-84.2010.
- [**Yang, 2010**] Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: J.R. González, D.A. Pelta, C. Cruz, G. Terrazas, N. Krasnogor (eds.) Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), vol. 284, Chap. 6, pp. 65–74. Springer, Berlin Heidelberg (2010)
- [**Yu et al, 2011**] Yu, Y., Jiang, C., Zhao, X., Yun, L., Li, Z., & Ren, Y. (2011). Sequence-based localization algorithm with improved correlation metric and dynamic centroid. Science China Information Sciences, 54(11), 2349. DOI <https://doi.org/10.1007/s11432-011-4434-1> "cit"
- [**Yu et Li, 2012**] W.Yu, and H. Li, "An Improved DV-Hop Localization Method in Wireless Sensor Networks," in Proceedings of Computer Science and Automation Engineering (CSAE), 2012, vol.3, pp. 199-202.

[Yu et Oppermann, 2004] Yu, K., & Oppermann, I. (2004, May). Performance of UWB position estimation based on time-of-arrival measurements. In *Ultra Wideband Systems, 2004. Joint with Conference on Ultrawideband Systems and Technologies. Joint UWBST & IWUWBS. 2004 International Workshop on* (pp. 400-404). IEEE. DOI: 10.1109/UWBST.2004.1321004

[Zhang et al, 2012] B. Zhang, M. Ji, and L. Shan, “A weighted centroid localization algorithm based on DV-hop for wireless sensor network,” in *Proceedings of the 8th International Conference on Wireless Communications, Networking and Mobile Computing*, Shanghai, China, 2012, pp. 1-5.

[Zhong, 2008] Zhong, J. W. W. L. (2008). Study on the Application of DV-Hop Localization Algorithms to Random Sensor Networks [J]. *Journal of Electronics & Information Technology*, 4(51).

Références bibliographiques

[Akyildiz et al, 2002] Ian F. Akyildiz ; W.Su ; Y.Sankarasubramaniam ; and E. Cayirci. A survey on sensor networks. volume 4, pages 393–422. IEEE Communications Magazine, Août 2002.

[Al Shayokh et Soo, 2017] Al Shayokh, Md, and Soo Young Shin. "Bio Inspired Distributed WSN Localization Based on Chicken Swarm Optimization." *Wireless Personal Communications* 97.4: 5691-5706.2017).

[Allaoua et Laoufi, 2009] Allaoua B. & Laoufi A., « Optimal Power Flow Solution Using Ant Manners for Electrical Network », *Advances in Electrical and Computer Engineering*, Vol. 9, N°. 1,2009.

[Amundson et Koutsoukos, 2009] I.Amundson ; X. D. Koutsoukos. A survey on localization for mobile wireless sensor networks. *International Workshop on Mobile Entity Localization (MELT)*, septembre 2009.

[Bahl et Padmanabhan, 2000] Bahl, P., & Padmanabhan, V. N. (2000). RADAR: An in-building RF-based user location and tracking system. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Vol. 2, pp. 775-784). DOI: 10.1109/INFCOM.2000.832252.

[Berthiau et Siarry, 2001] Berthiau, Gerard & Siarry, Patrick. (2001). État de l'art des méthodes "d'optimisation globale". *RAIRO - Operations Research*. 35. 329 - 365. 10.1051/ro:2001118.

[Biswas et Ye, 2004] Biswas, P., & Ye, Y. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks* (pp. 46-54). ACM. (2004, April).

[Bontempi, 2006] G. Bontempi. *Statistical Foundations of Machine Learning*. Université Libre de Bruxelles, 2006.

[Boukerche, 2008] Azzedine Boukerche, —*Algorithms And Protocols For Wireless Sensor Networks*], Wiley, ISBN: 978-0-471-79813-2, 11-2008.

[Boushaba et al, 2009] Boushaba, M., Hafid, A., & Benslimane, A. (2009). High accuracy localization method using AoA in sensor networks. *Computer Networks*, 53(18), 3076-3088. DOI:10.1016/j.comnet.2009.07.015.

[Brown et al, 2007] Brown, C., Liebovitch, L. S., & Glendon, R. (2007). Lévy flights in DobeJu/'hoansi foraging patterns. *Human Ecology*, 35, 129–138.

[Bulusu et al, 2000] Bulusu, N., Heidemann, J., Estrin, D.:GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, vol. 7, no. 5, Oct. 2000, pp. 28-34

[Cao et al, 2012] Cao, S., Wang, J., & Gu, X. A wireless sensor network location algorithm based on firefly algorithm. In *AsiaSim 2012* (pp. 18-26). Springer, Berlin, Heidelberg. (2012).

- [**Cardei et al, 2005**] Mihaela Cardei, Ionut Cardei, Ding-Zhu Du, —Resource Management in Wireless Networking, Springer, ISBN: 978-0-387-23807-4, 01-2005
- [**Černý , 1985**] V. Černý. A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41-51, 1985
- [**CHALLAL, 2008**] CHALLAL. Yacine. Réseaux de Capteurs Sans Fils : Support de cours. Ecole nationale Supérieure d'Informatique, Algérie. Novembre 2008. <https://moodle.utc.fr/file.php/498/support-SIT60.pdf>
- [**Chen et al, 2008**] H.Chen, K.Sezaki, P.Deng, and H.C.So, "An improved DV-hop localization algorithm for wireless sensor networks," in *Proceedings of IEEE Conference on Industrial Electronics and Applications*, Singapore, 2008, pp.1557-1561.
- [**Cho et al, 2011**] Cho, H. H., Lee, R. H., & Park, J. G. (2011). Adaptive parameter estimation method for wireless localization using RSSI measurements. *Journal of Electrical Engineering and Technology*, 6(6), 883-887. DOI:10.1007/978-3-642-22309-9_29
- [**Cuevas et al, 2010**] Cuevas.E, Zaldivar.D ,Valentine.M. A novel multi-threshold segmentation approach based on differential evolution optimization. *Expert systems with application* Vol.37, pp.5265-5271.2010.
- [**Dargie et Poellabauer, 2010**] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*, John Wiley & Sons, 2010.
- [**Deng et al, 2008**] B.Deng H.Liu L.Zhang, L.Duan and G.Huang. S-mrl distributed localization algorithm for wireless sensor networks. *Wireless Communications, Networking and Mobile Computing*, 2008. WiCOM '08. 4th International Conference on, pages 1–4, 2008.
- [**Dhurandher et al, 2008**] S. Dhurandher, S. Misra, M. S. Obaidat and S. Khairwal, "UWSim: a simulator for underwater wireless sensor networks," *Simulation: Transactions of the Society for Modeling and Simulation International, SCS*, Vol. 84, No. 7, pp. 327–338, July 2008.
- [**Ding et al, 2012**] Ding, J., Zhang, L., Cheng, G., Ling, Z., Zhang, Z., & Lei, Y. (2012). Study on DV-Hop algorithm based on modifying hop count for wireless sensor networks. *International Journal of Computer Science Engineering & Technology* . Vol. 2 Issue 10, p1452-1456. 5p.
- [**Dorigo et Caro, 1999**] Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on (Vol. 2, pp. 1470-1477)*. IEEE. DOI: 10.1109/CEC.1999.782657.
- [**Dorigo et Stutzle, 2004**] M. Dorigo And T. Stützle. *Ant Colony Optimization*, Cambridge: MIT Press, 2004.
- [**Dorigo, 1992**] Dorigo, M.: *Optimization, learning and natural Algorithms*. Ph.D. thesis, Politecnico di Milano, Italy (1992)
- [**Dorigo, 2007**] M. Dorigo. *Ant Colony Optimization*, Scholarpedia, Vol.2, N°03,ppn 1461.http://www.scholarpedia.org/article/Ant_colony_optimization, 2007.

[Dreo and Siarry, 2003] Dreo, J. and P. Siarry. Diverses techniques d'optimisation inspirées de la théorie de l'auto-organisation dans les systèmes biologiques. In . Paris, France. 2003.

[Farmer et al, 1992] Farmer, J.D., Packard, N.H., Perelson, A.S.: The immune system, adaptation, and machine learning. *Phys. D* 2(1–3), 187–204 (1986)

[Feo et Resende, 1989] Feo, T.A., Resende, M.G.C.: A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters* 8(2), 67–71 (1989)

[Feyel, 2015] Philippe Feyel. Optimisation des correcteurs par les métaheuristiques. Application à la stabilisation inertielle de ligne de visée. Autre. CentraleSupélec, 2015. Français. NNT : 2015CSUP0011. tel-01260457v2.

[Fonseca et Fleming, 1993] Fonseca, C. M., & Fleming, P. J. (1993, June). Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization. In *Icga* (Vol. 93, No. July, pp. 416-423).

[Geem et al, 2001] Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: Harmony search. *Simulation* 76(2), 60–68 (2001)

[Girod et al, 2002] Girod, L., Bychkovskiy, V., Elson, J., & Estrin, D. (2002). Locating tiny sensors in time and space: A case study. In *Computer Design: VLSI in Computers and Processors, 2002. Proceedings. 2002 IEEE International Conference on* (pp. 214-219). IEEE. .DOI: 10.1109/ICCD.2002.1106773.

[Glover, 1986] Glover, F.: Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* 13(5), 533–549 (1986)

[Goldberg, 1989] Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1st edn. Studies in Computational Intelligence. Addison-Wesley Longman (1989)

[Gopakumar et Jacob, 2008] Gopakumar, A., & Jacob, L. Localization in wireless sensor networks using particle swarm optimization. (2008).

[Goyal et Patterh, 2014] Goyal, S., & Patterh, M. S. Wireless sensor network localization based on cuckoo search algorithm. *Wireless personal communications*, 79(1), 223-234. (2014).

[Gui et al, 2015] L.Gui , T. Val, A. Wei, and R. Dalce,” Improvement of range-free localization technology by a novel DV-hop protocol in wireless sensor networks”, *Ad Hoc Networks*, vol. 24, pp.55-73, 2015.

[Hall, 1999] Prentice Hall, T.S. Rappaport. *Wireless Communications, Principles & Practices*1999.

[Hansen et al, 1995] Hansen, N., Ostermeier, A., Gawelczyk, A.: On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In: *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 57–64. Morgan Kaufmann, San Francisco (1995).

[Harter et al, 2002] A. Harter, A. Hopper, P. Steggles, A. Ward and P. Webster, 2002. The anatomy of a context-aware application. *Wireless Networks*, 8: pp. 187-197. <https://doi.org/10.1023/A:1013767926256>.

[He et al, 2003] He, T., Huang, C., Blum, B. M., Stankovic, J. A., & Abdelzaher, T. (2003, September). Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*(pp. 81-95). ACM. Doi:10.1145/938985.938995.

[He et al, 2006] He, S., Wu, Q., Saunders, J.: A novel group search optimizer inspired by animal behavioural ecology. In: *Proceedings of 2006 IEEE Congress on Evolutionary Computation*, pp. 16–21, Vancouver (2006).

[Hightower et, Borriello 2001] Jeffrey Hightower and Gaetano Borriello, *Location Systems for Ubiquitous Computing*, IEEE Computer Journal, VOL. 34 N° 8, pp. 57-66, 2001.

[Hofmeyr et Forrest, 2000] Hofmeyr, S. A., & Forrest, S. (2000). Architecture for an artificial immune system. *Evolutionary computation*, 8(4), 443-473. DOI:10.1162/106365600568257.

[Holland, 1975] Holland. J. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor. 1975.

[Huang et al, 2003] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. ACM Press New York, NY, USA, 2003.

[Ji et Zha, 2004] Ji, X., & Zha, H. (2004, March). Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies* (Vol. 4, pp. 2652-2661). IEEE. DOI: 10.1109/INFCOM.2004.1354684.

[Kannan et al, 2006] Kannan, A. A., Mao, G., & Vucetic, B. Simulated annealing based wireless sensor network localization. *JCP*, 1(2), 15-22. (2006).

[KARA, 2009] Messaoud KARA, Réseau de capteurs sans fil : Etude en vue de la réalisation d'un récepteur GPS différentiel à faible coût. THÈSE de doctorat; Université Blaise Pascal – Clermont-Ferrand II. 2009.

[KARABOGA et BASTURG, 2007] KARABOGA D., BASTURG B., « A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm », *Journal of Global Optimization*, Springer, Vol. 39, Issue 3, p. 459-471, Avril 2007.

[Karaboga et Basturg, 2008] KARABOGA D., BASTURG B., « On the performance of artificial bee colony (ABC) algorithm », *Journal Applied Soft Computing*, Elsevier, Vol. 8, Issue 1, p.687-697, Janvier 2008.

[Karl et Willig, 2005] Holger Karl, Andreas Willig, —*Protocols And Architectures For Wireless Sensor Networks*, Wiley, ISBN: 978-0-470-09510-2, 04-2005

[Kaur et al, 2017] Kaur, Amanpreet & Gupta, Govind & Kumar, Padam. (2017). A Survey of Recent Developments in DV-Hop Localization Techniques for Wireless Sensor

Network. Journal of Telecommunication, Electronic and Computer Engineering (JTEC). 9. 61-71.

[**Kennedy et Eberhart, 1995**] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995).

[**Kirkpatrick et al, 1983**] Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. *Science* 220(4598), 671–680 (1983)

[**Koza, 1992**] Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems), 1st edn. MIT Press (1992).

[**Kulaib et al, 2011**] A. R. Kulaib, R. M. Shubair, M. A. Al-Qutayri, and J. W. P. Ng, ‘An overview of localization techniques for wireless sensor networks,’ in Proc. Int. Conf. Innov. Inf. Technol. (IIT), Apr. 25–27, 2011, pp. 167–172.

[**Laptik et Navakauskas, 2007**] Laptik.R, Navakauskas.D. Application of Ant Colony Optimization for Image Segmentation. *Electronics and Electrical Engineering*. Kaunas: Technologija, Vol.80. No. 8. pp. 13–18.2007.

[**Li et Kunz, 2007**] Li, L., & Kunz, T. (2007, October). Localization applying an efficient neural network mapping. In Proceedings of the 1st international conference on Autonomic computing and communication systems (p. 1). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). <http://dx.doi.org/10.4108/ICST.AUTONOMICS2007.2126>.

[**Lin et al, 2005**] K. Lin, J. Yu, J. Hsu, S. Zahedi, D. Lee, J. Friedman, A. Kansal, V. Raghunathan, and M. Srivastava. ‘Helio mote: enabling long-lived sensor networks through solar energy harvesting’. In SenSys ‘05: Proceedings of the 3rd international conference on Embedded networked sensor systems, pages 309_309, New York, NY, USA, 2005.ACM.

[**MacArthur et Wilson, 1967**] MacArthur, R., Wilson, E.: The Theory of Biogeography. Princeton University Press, Princeton, NJ (1967).

[**Marks et Niewiadomska, 2007**] Marks, M., & Niewiadomska-Szynkiewicz, E. Two-phase stochastic optimization to sensor network localization. In Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on (pp. 134-139). IEEE. (2007, October).

[**Meng et al, 2014**] Meng, X., Liu, Y., Gao, X., & Zhang, H. (2014). A new bio-inspired algorithm: Chicken swarm optimization. In Advances in swarm intelligence (pp. 86–94). Springer, Berlin.

[**Metropolis et al, 1953**] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. "Equations of State Calculations by Fast Computing Machines". *Journal of Chemical Physics*, 21(6):1087- 1092, 1953.

[**Moussa et El-Sheimy, 2010**] A. Moussa and N. El-Sheimy, "Localization of wireless sensor network using Bees Optimization Algorithm," The 10th IEEE International Symposium on Signal Processing and Information Technology, Luxor, 2010, pp. 478-481.

[**Mühlenbein et Paaß, 1996**] Mühlenbein, H., Paaß, G.: From recombination of genes to the estimation of distributions. I.Binary parameters. In: Proceedings of the 4th International

Conference on Parallel Problem Solving from Nature, PPSN IV, pp. 178–187. Springer, London (1996).

[**Nan et al, 2007**] Nan, G. F., Li, M. Q., & Li, J. Estimation of node localization with a real-coded genetic algorithm in WSNs. In Machine learning and cybernetics, 2007 International Conference on (Vol. 2, pp. 873-878). IEEE. (2007, August).

[**Nicolescu et Nath, 2001**] Nicolescu D, Nath B. "Ad-Hoc positioning systems (APS)". Proceedings of the 2001 IEEE Global Communications Conference (GCC'01). New York USA: IEEE, 2001, pp. 2926-2931.

[**Nicopolitidis et al, 2003**] P. Nicopolitidis, M. S. Obaidat, G. I. Papadimitriou and A. S. Pomportsis, Wireless Networks, John Wiley & Sons, 2003.

[**Niculescu et Nath, 2003**] Niculescu, D., & Nath, B. (2003). DV based positioning in ad hoc networks. Telecommunication Systems, 22(1-4), 267-280. DOI:10.1023/A:1023403323460

[**Obaidat et al, 2009**] M. S. Obaidat, P. Nicopolitidis and J.-S. Li, "Security in wireless sensor networks," Security and Communications Networks, Wiley, Vol. 2, No. 2, pp. 101–103, March/April 2009.

[**Onwubolu et Babu, 2004**] Onwubolu G. C., and Babu B. V. New Optimization Techniques in Engineering, Springer-Verlag, Germany.2004.

[**Pan, 2011**] Pan, W. T. (2011). A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowledge-Based Systems. 2011, 26. pp: 69-74. <https://doi.org/10.1016/j.knosys.2011.07.001>

[**Passino, 2002**] Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine 22(3), 52–67 (2002). doi:10.1109/MCS.2002.1004010

[**Paul et Sato, 2017**] Paul, A. K., & Sato, T. (2017). Localization in Wireless Sensor Networks: A Survey on Algorithms, Measurement Techniques, Applications and Challenges. Journal of Sensor and Actuator Networks, 6(4), 24. <https://doi.org/10.3390/jsan6040024>

[**Pavlyukevich, 2007**] Pavlyukevich, I. (2007). Cooling down Lévy flights. Journal of Physics A, Mathematical and Theoretical, 40, 12299–12313.

[**Peng et Li, 2015**] B. Peng, and L. Li, "An improved localization algorithm based on genetic algorithm in wireless sensor networks", Cognitive Neurodynamics, vol. 9, no.2, pp. 249-256, 2015.

[**Price et al., 2005**] Price V. Kenneth., Storn M. Rainer., and Lampinen A. Jouni. Differential evolution: A practical approach to global optimization. Springer-Verlag Berlin, Heidelberg.2005.

[**Rabhi et Fizazi, 2015**] S.Rabhi, H.Fizazi. Étude comparative entre l’algorithme de firefly et les comportements collectifs des fourmis dans la classification des images satellitaire. Troisième conférence sur la vision artificielle CVA 2015. Tizi ousou. Algérie. 2015.

- [**Raghunathan et al, 2002**] V. Raghunathan ; C. Schurgers ; S. Park ; and M. B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, Volume: 19 , Issue: 2 , Mar 2002.
- [**Rashedi et al, 2009**] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: A gravitational search algorithm. *Information Sciences* 179(13), 2232–2248 (2009). doi:10.1016/j.ins.2009.03.004
- [**Saad, 2008**] Clément Saad. Quelques contributions dans les réseaux de capteurs sans fil : Localisation et Routage. *Réseaux et télécommunications [cs.NI]*. Thèse de doctorat, Université d'Avignon, 2008. Français.
- [**Safa, 2014**] H.Safa, “A novel localization algorithm for large scale wireless sensor networks”, *Computer Communications*, vol. 45, pp 32-46, 2014.
- [**Sassi et al, 2014**] Sassi, H., Najeh, T., & Liouane, N. (2014). A Selective 3-Anchor DV-Hop Algorithm Based On the nearest anchor for wireless sensor network. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 8(10), 1756-1760.
- [**Savvides et al, 2002**] A. Savvides, H. Park, et M. Srivastava, 2002. The bits and flops of the n-hop multilateration primitive for node localization problems. *First ACM International WSNA*, Atlanta.
- [**Shlesinger et Zaslavsky, 1995**] Shlesinger, M. F., Zaslavsky, G.M., & Frisch, U. (Eds.). (1995). *Lévy flights and related topics in physics*. Berlin: Springer.
- [**Shlesinger, 2006**] Shlesinger, M. F. (2006). Search research. *Nature*, 443, 281–282.
- [**Simic et Sastry, 2003**] S. Simic and S. Sastry. A distributed algorithm for localization in random wireless networks. *Discrete Applied Mathematics*, 2003.
- [**Simon, 2008**] Simon, D.: Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation* 12(6), 702–713 (2008)
- [**Sivakumar et Venkatesan, 2016**] S. Sivakumar & Raman Venkatesan. Error Minimization in Localization of Wireless Sensor Networks using Ant Colony Optimization. *International Journal of Computer Applications* 145(8):15-21 (July 2016)
- [**Sohraby et al, 2007**] K. Sohraby, D. Minoli and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*, John Wiley & Sons, 2007.
- [**Song et Tam, 2015**] G. Song, and D. Tam,” Two Novel DV-Hop Localization Algorithms for Randomly Deployed Wireless Sensor Networks”, *International Journal of Distributed Sensor Networks*, 2015.
- [**Srikanth et al, 2018**] Banda Srikanth, Hemant Kumar, Karanam U.M Rao (2018). A robust approach for WSN localization for underground coal mine monitoring using improved RSSI technique. *Mathematical Modelling of Engineering Problems*. vol. 5, No. 3, September, 2018, pp. 225-231 DOI:10.18280/mmep.050314
- [**Stojmenovic, 2005**] Ivan Stojmenovic, —*Handbook Of Sensor Networks: Algorithms And Architectures*], Wiley-Intersciencene, ISBN: 978-0-471-68472-5, 10-2005

- [**Storn et Price, 1997**] R. Storn and Price. K. Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* Vol.11, pp 341–359, 1997.
- [**Storn et Price, 1997**] Storn, R.M., Price, K.V.: Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11(4), 341–359 (1997)
- [**Su et al, 2001**] Y. Sankarasubramaniam E. Cayirci I.F. Akyildiz, W. Su*. Wireless sensor networks: a survey. volume 4, pages 393–422, December 2001.
- [**Tomic et Mezei, 2016**] S. Tomic, and I. Mezei, "Improvements of DV-Hop localization algorithm for wireless sensor networks." *Telecommunication Systems*, pp. 1-14, 2016.
- [**Törn et Zilinskas, 1989**] A. Törn et A. Zilinskas, *Global optimization*, 'edit' e par G. Goos et J. Hartmanis. Springer Verlag, No. 350 (1989).
- [**Walker et al, 1993**] Walker, A., Hallam, J., Willshaw, D.: Bee-havior in a mobile robot: The construction of a self-organized cognitive map and its use in robot navigation within a complex, natural environment. In: *Proceedings of ICNN'93, International Conference on Neural Networks*,
- [**WANG et al, 2010**] WANG, Dianhong, JIA, Hongdong, CHEN, Fenxiong, et al. An improved DV-Distance localization algorithm for wireless sensor networks. In : 2010 2nd International Conference on Advanced Computer Control. IEEE, 2010. p. 472-476.
- [**Yang , 2008**] Yang X. S. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.
- [**Yang et Deb, 2009**] Yang, X., & Deb, S. (2009). Cuckoo search via Lévy flights. In *World congress on nature & biologically inspired computing (NaBIC2009)*, IEEE, 978-1-4244-5612-3/09.
- [**Yang, 2009**] Yang X.S, Firefly algorithm for multimodal optimization in proceedings of the stochastic Algorithms, SAGA 109. Vol.5792, Oct.2009.
- [**Yang, 2010**] Yang, X. S. Firefly Algorithm, Stochastic Test Functions and Design Optimisation . *Int. J. Bio-Inspired Computation*, Vol. 2, No. 2, pp.78-84.2010.
- [**Yang, 2010**] Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: J.R. González, D.A. Pelta, C. Cruz, G. Terrazas, N. Krasnogor (eds.) *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, vol. 284, Chap. 6, pp. 65–74. Springer, Berlin Heidelberg (2010)
- [**Yu et al, 2011**] Yu, Y., Jiang, C., Zhao, X., Yun, L., Li, Z., & Ren, Y. (2011). Sequence-based localization algorithm with improved correlation metric and dynamic centroid. *Science China Information Sciences*, 54(11), 2349. DOI <https://doi.org/10.1007/s11432-011-4434-1> "cit"
- [**Yu et Li, 2012**] W.Yu, and H. Li, "An Improved DV-Hop Localization Method in Wireless Sensor Networks," in *Proceedings of Computer Science and Automation Engineering (CSAE)*, 2012, vol.3, pp. 199-202.

[Yu et Oppermann, 2004] Yu, K., & Oppermann, I. (2004, May). Performance of UWB position estimation based on time-of-arrival measurements. In Ultra Wideband Systems, 2004. Joint with Conference on Ultrawideband Systems and Technologies. Joint UWBST & IWUWBS. 2004 International Workshop on (pp. 400-404). IEEE. DOI: 10.1109/UWBST.2004.1321004

[Zhang et al, 2012] B. Zhang, M. Ji, and L. Shan, "A weighted centroid localization algorithm based on DV-hop for wireless sensor network," in Proceedings of the 8th International Conference on Wireless Communications, Networking and Mobile Computing, Shanghai, China, 2012, pp. 1-5.

[Zhong, 2008] Zhong, J. W. W. L. (2008). Study on the Application of DV-Hop Localization Algorithms to Random Sensor Networks [J]. Journal of Electronics & Information Technology, 4(51).