République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Ahmed Draia - Adrar

Faculté des Sciences et de la Technologie

Département des Mathématiques et Informatique

Mémoire de fin d'étude, en vue de l'obtention du diplôme de Master en informatique

Option : Systèmes Intelligents

Thème

Réseaux de capteurs sans fil: Problèmes de localisation

Préparé par

BENABDERRAHMANE Abderrahmane et HANTAOUI Abdelkarim

Encadré par

Mr. KADDI Mohammed

Année Universitaire 2019/2020

Résumé

La connaissance des positions des senseurs dans l'environnement est souvent souhaitable, afin de pouvoir déterminer l'origine des flux de mesures collectées. L'algorithme de localisation proposé dans ce mémoire a pour but d'estimer ces positions de manière automatique en utilisant une méthode requiert l'encodage au préalable de la position d'un certain nombre d'ancres. L'algorithme BATDV-Hop proposé basé sur les deux algorithmes: algorithme DV-Hop et algorithme de chauves-souris. Le langage MATLAB a été utilisé pour tester la performance de la technique suggérée.

Mots clé:

Réseaux de capteurs sans fil, localisation, algorithme DV-Hop, algorithme de chauves-souris, BATDV-Hop.

Abstract

Knowledge of the positions of the sensors in the environment is often desirable, in order to be able to determine the origin of the flow of collected measurements. The purpose of the localization algorithm proposed in this memory is to estimate these positions automatically by using a method requiring the prior encoding of the position of a certain number of anchors. The proposed BATDV-Hop algorithm based on the two algorithms: DV-Hop algorithm and BAT algorithm. MATLAB language was used to test the performance of the suggested technique.

Keywords:

Wireless sensor networks, localization, DV-Hop algorithm, BAT algorithm, BATDV-Hop.

الملخص:

غالبًا ما تكون معرفة مواقع أجهزة الاستشعار في البيئة أمرًا مرغوبًا فيه ، من أجل التمكن من تحديد أصل تدفق القياسات المجمعة. الغرض من خوارزمية تحديد الموقع المقترحة في هذه المذكرة هو تقدير هذه المواضع تلقائيًا باستخدام طريقة تتطلب تشفيرًا مسبقًا لموضع عدد معين من المراسي. تعتمد خوارزمية BATDV-Hop المقترحة على خوارزميتين: خوارزمية DV-Hop وخوارزمية BAT .تم استخدام لغة الماتلاب لاختبار أداء التقنية المقترحة.

الكلمات المفتاحية:

شبكات الاستشعار اللاسلكية ، تحديد الموقع، خوارزمية DV-Hop ، خوارزمية الخفافيش (BAT)، BATDV-Hop .

Remerciement

Tout d'abord, nous remercions le bon DIEU, notre créateur de nous avoir donné la force, la volonté et le courage afin d'accomplir ce travail modeste.
nous adressons le grand remerciement à notre encadreur ***Mr: KADDI Mohammed*** *qui a proposé le thème de ce mémoire, pour ses conseils et ses dirigés du début à la fin de ce travail.*

nous voudrais également remercier les membres de jurys pour avoir accepté d'évaluer ce travail et pour toutes leurs remarques et critiques, ainsi et son aide et son encouragement.
Tout le personnel et les enseignants du notre département pour leur soutien inestimable.
A tous mon enseignants pour leur conseils.
Sans oublier bien sûr de remercier profondément tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Mr.BENABDERRAHMANE et Mr.HANTAOUI Abdelkarim

Dédicace

Je dédie ce modeste travail :
A ma mère, à mon père
À mes proches frères et ma sœur,
chacun par
son nom.
À toute ma famille.
À tous mes amis.
À tous mes chers enseignants..
À mon binôme Abdelkraim.
A tous les étudiants de la promotion 2019/2020
Option : système intelligent

Abderrahmane BENABDERRAHMANE

Table des matières

Table des matières

Table des matières

Table des Figures

Liste des tableaux

Table des acronymes

RCSFs	Réseaux de capteurs sans fils
GPS	Global Positioning System
IRNSS	Indian Regional Navigational Satellite System
DV-hop	Distance vecteur hop
CO-WVSS	Correct and optimize the weight value from similitude signal
FDV-Hop	Feedback distance vecteur
GGDI	grands ligne de direction de gain d'intersection
TOA	Time of Arrival.
CPE	Common platform enuumeration
APS	Access points
RSSI	Received signal strength indication
MDS	multidimensional
BA	Bat Algorithme
BDV-Hop	Bat distance vecteur
AOA	Angle of Arrival
WSN	Wireless sensor Networks
LEACH	LOW énergie adaptive clustering hierarchy
PSO	Particle swarm optimization

Introduction général

Au cours des dernières décennies, de nombreuses études ont porté sur un nouveau domaine de recherche tel que le réseau de capteurs sans fil, en raison de son rôle important qui répond aux besoins de nombreuses applications industrielles et scientifiques. Ce réseau est utilisé dans différents domaines qu'ils soient médicales, environnementaux, militaires ou encore sécuritaires...etc.

Cependant, un capteur est un équipement de taille très réduite, englobant des ressources très limitées en matière de mémoire, de calcul et alimenté avec des batteries de faible puissance. Ces caractéristiques spécifiques impliquent de nouveaux défis lors de la mise en œuvre d'un réseau pareil. Plusieurs challenges ont été identifiés par les chercheurs incluant la découverte du réseau, le routage et le contrôle du réseau Le traitement collaboratif des informations, la localisation, la collecte et la dissémination des données.

Dans ce mémoire, nous allons intéresser aux problèmes relatifs à la localisation dans les réseaux de capteurs sans fil. Les techniques de localisation classées en deux grand classe selon l'utilisation ou pas des nœud spécifique dite ancres (les nœuds qui nous connaissons leur emplacement depuis le début). Nous nous sommes intéressés aux approche sans ancres, on a implémenté l'algorithme de DV-Hop afin d'évaluer ses performances et nous essayons d'améliorer la précision de cet algorithme par l'hybridation avec un heuristique performant de voisinage appelés la chauves-souris.

La suite de ce document est constituée en les quatre chapitres suivant:

Chapitre 1 : Généralités sur les réseaux de capteurs sans fil .

Chapitre 2: Localisation dans les RCSFS

Chapitre 3: L'Algorithme des chauves-souris

Chapitre 4 : Simulation. BATDV-Hop

A la fin de ce mémoire, une conclusion est donnée pour résumer notre travail.

Chapitre 1

Généralités sur les réseaux de capteurs sans fil

Sommaire

1.1 Introduction

Les développements récents des technologies micro-capteurs et des communications sans fils ont engendré l'apparition des réseaux de capteurs sans fils (RCSF ou Wireless Sensor Nete-work).Un tel réseau est constitué d'un grand nombre de dispositifs physiques appelés capteur ou nœud capables de relever, de traiter et de transmettre des informations de l'environnement, dans lequel ils sont déployés, à un où plusieurs points de collecte[1]. L'un des problèmes ma-jeurs de ce type de réseaux est la localisation qui consiste à déterminer l'emplacement des nœuds de capteur dans une zone géographique.

Dans ce chapitre, nous présenterons les réseaux de capteurs sans fils et ce en décrivant leur architecture, leurs caractéristiques, leurs domaines d'application, les contraintes de conception d'un tel type de réseau.

1.2 Définition d'un capteur

Un capteur est un petit appareil autonome capable d'effectuer de simples mesures sur son environnement immédiat, telles que la température, les vibrations et la pression. Cet appareil est augmenté de capacités de calcul et de communication ainsi que de batteries lui conférant une autonomie [2].

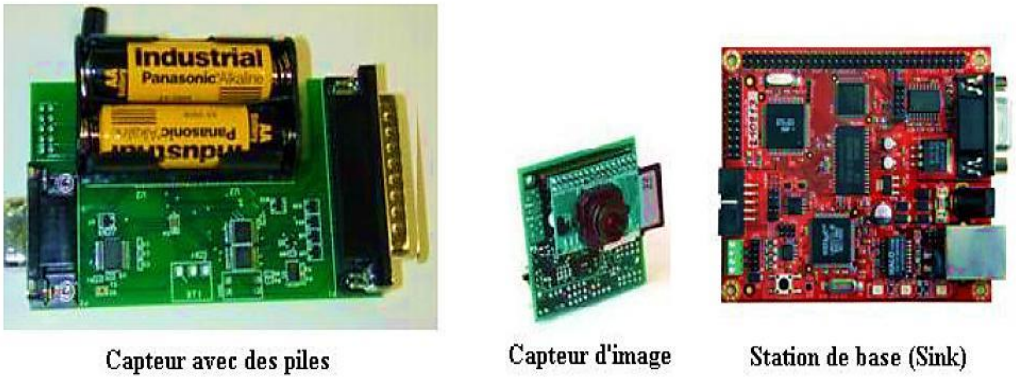

Figure 1.1: Quelques capteurs existants sur le march

1.2.1 Types de capteurs

Les capteurs sont disponibles en déférents modèles et dépendent de l'application à laquelle ils sont destinés. Il existe plusieurs fabricants de capteurs parmi lesquels nous trouvons : Imote IV, Art of Technologie et Crossbow [3]. Les déférents capteurs : MICA2, Telos B, MICAz, Imote2, etc.

Figure 1. 2: Symbole de la compagnie Crossbow

.

Figure 1. 3: Capteur MicaZ

1.3 Réseaux de capteurs sans fil (RCSF)

Un réseau de capteurs est constitué d'un grand nombre d'unités appelées nœuds capteurs. Chaque nœud est composé principalement d'un ou plusieurs capteurs. Il est capable de surveiller son environnement et de réagir en cas de besoin en envoyant l'information collectée à un ou plusieurs points de collecte. Ces nœuds communiquent entre eux selon une topologie (fixe ou mobile) du réseau afin d'acheminer les informations à une unité de commande en dehors de la zone de mesure.

1.3.1 Nœud capteur

Un capteur est un dispositif ayant pour tâche de transformer une mesure physique observée en une mesure généralement électrique qui sera à son tour traduite en une donnée binaire exploitable et compréhensible par un système d'information [4].

Parmi les différents types de mesures enregistrées par les capteurs, on peut citer entre autres : la température, l'humidité, la luminosité, l'accélération, la distance, les mouvements, la pression, la présence d'un gaz, la vision (capture d'image), le son, etc.

Les nœuds de capteurs ont des composants matériels et logiciels.

1.3.2 Composants logiciels

La contrainte énergétique des capteurs exige l'utilisation de systèmes d'exploitation légers tels que TinyOS [4] (Open Source) et son langage de programmation NesC, Contiki et MantisOS [4] Cependant, TinyOS reste toujours le plus utilisé et le plus populaire dans le domaine des RCSF. Il est libre et est utilisé par une large communauté de scientifiques dans des Simulations pour le développement et le test des algorithmes et protocoles réseau [4].

1.3.3 Composants matériels

- **Unité d'acquisition**

L'unité d'acquisition est composée d'un ou plusieurs capteurs qui vont obtenir des mesures numériques sur les paramètres environnementaux (température, pression, humidité, le bruit, les vibrations et les changements du paramètre d'état de la personne, par exemple la pression artérielle et le rythme cardiaque) et d'un convertisseur Analogique/Numérique (CAN) qui va convertir l'information relevée et la transmettre à l'unité de traitement **[4]**.

- **Unité de traitement**

L'unité de traitement est composée de deux interfaces, une interface pour l'unité d'acquisition et une interface pour l'unité de transmission. Cette unité est également composée d'un processeur et d'un système d'exploitation spécifique. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de transmission**[4]**.

- **Unité de transmission**

L'unité de transmission est responsable de toutes les émissions et réceptions de données via un support de communication radio.

- **Unité d'énergie**

C'est la batterie qui, n'est généralement ni rechargeable ni remplaçable. La capacité d'énergie limitée au niveau des capteurs représente la contrainte principale lors de la conception de protocoles pour les réseaux de capteurs**[4]**.

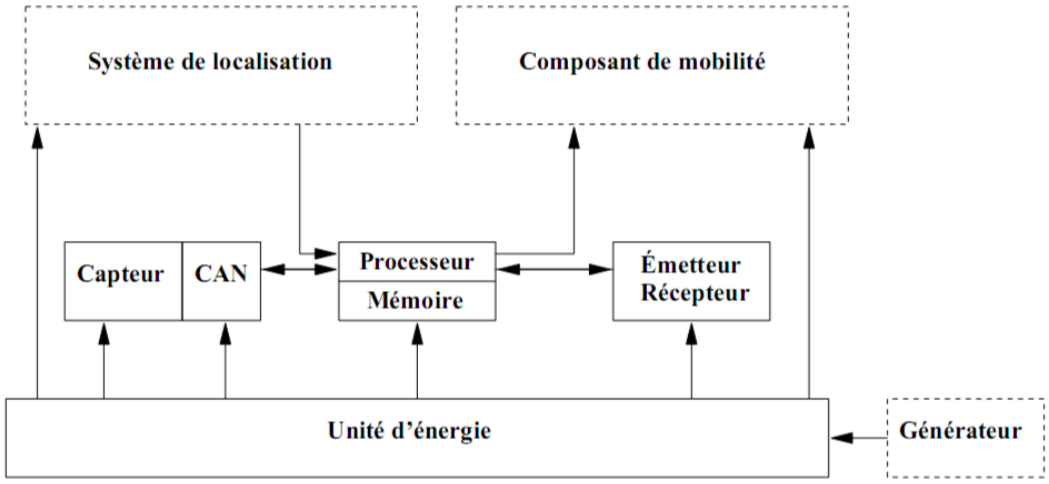

Figure 1.4 : Composants d'un capteur [1]

D'autres composants peuvent être ajoutés à un nœud capteur, comme un système de localisation (GPS : *Global Positioning System* qui est un système de géo localisation fonctionnant au niveau mondial et reposant sur l'exploitation de signaux radio émis par des satellites dédiés)**,** un composant de mobilité pour le rendre mobile, un générateur d'énergie, etc.

Les principaux avantages des nœuds capteurs sont : leur taille réduite voir figure1.2, leur très faible consommation électrique et surtout leur capacité à communiquer sans fil (ce qui permet une grande liberté de mouvement par rapport aux nœuds filaires).

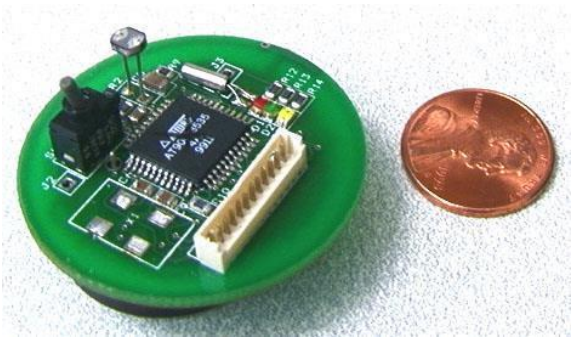

Figure 1.5 : Exemple d'un capteur

1.4 Architecture et fonctionnement d'un réseau de capteurs

Les réseaux de capteurs sans fil sont constitués de milliers des nœuds capteurs (sensors), ils sont considérés comme un type spécial de réseaux ad-Hoc [5] (figure 1.3). Ils apportent une perspective intéressante : celle de réseaux capables de s'auto-configurer et de s'autogérer sans qu'il y ait besoin d'interventions humaines. Les nœuds sont généralement déployés de manière aléatoire à travers une zone géographique, appelée zone d'intérêt *« champ de captage »*.

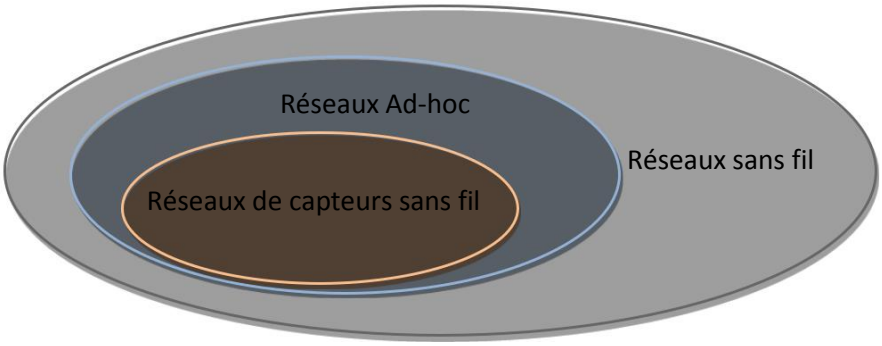

Figure 1.6: Réseaux de capteurs sans fil vs Réseaux Ad-hoc [5]

Un réseau de capteurs sans fil permettant de capter et collecter des événements, d'analyser les traitements et de transmettre les informations recueillies dans différents environnements .Les données récoltées sont acheminées grâce à des communications sans fil en multi-saut (c.-à-d. de proche en proche) à une station de base (ou puits « *sink* »), Ces nœuds-puits sont des bases de contrôle qui possèdent plus de ressources matérielles et dont le rôle est d'agréger/exploiter les données récoltées issues des capteurs. Elle représente en quelque sorte le point d'entrée du réseau de capteurs [5]. **(Figure 1.6).**

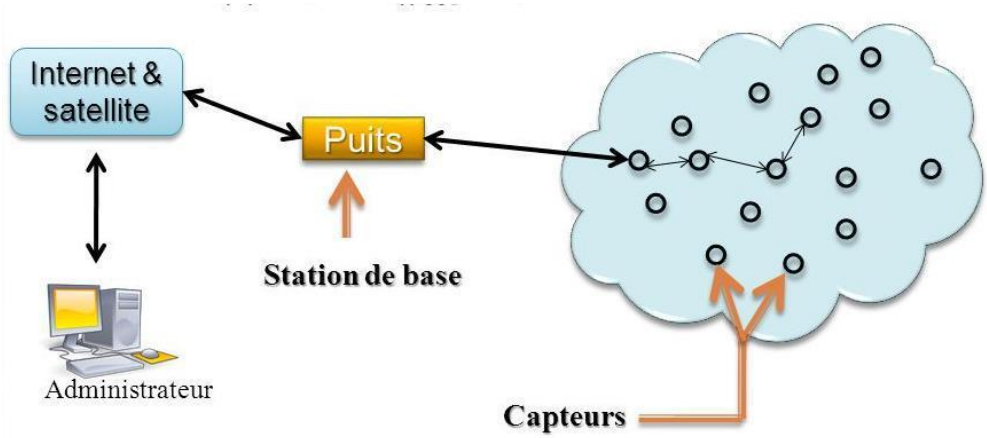

Figure 1.7: Architecture de la communication d'un réseau RCSF

En d'autres termes le fonctionnement d'un réseau de capteurs se déroule de la manière suivante : Les nœuds sont déployés dans une zone de captage pour la surveiller.

a) En cas d'un événement

Lorsqu'un nœud détecte un évènement (changement brusque de température, mouvement...), il le traite localement et l'achemine vers la station de base via une communication multi-saut [5] (figure1.5).

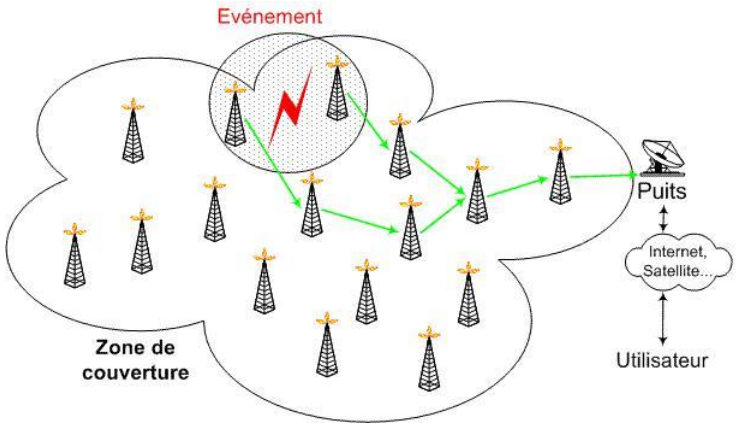

Figure1.8: Acheminement d'un événement [5]

b) En cas d'une demande

Lorsque l'on souhaite avoir l'état de la zone de couverture à un moment T, le puits émet des brodcasts vers toute la zone pour que les capteurs remontent leur dernier relevé vers le puits. Les informations sont alors acheminées par une communication multi-sauts [5] (figure1.6).

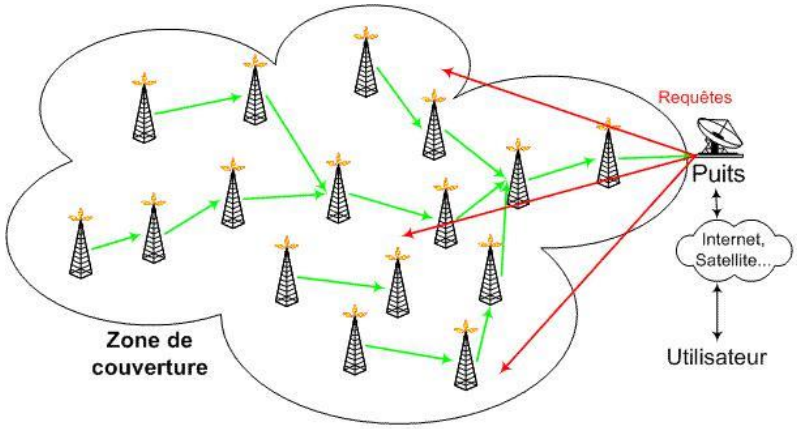

Figure 1.9: Acheminement d'une demande [5]

Dans les deux cas, le puits récupère les informations remontées par les différents capteurs et les transmet au centre de traitement (utilisateur final) à travers un réseau de communication, éventuellement l'Internet.

1.5 Applications des réseaux de capteurs

Le concept de réseaux de capteurs sans fil est basé sur une simple équation [6]« Capteurs + Processeur + Radio = Une centaine d'applications potentielles »

La taille de plus en plus réduite des micro-capteurs, le coût de plus en plus faible, la large gamme des types de capteurs disponibles (thermique, optique, vibrations,...) ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs d'envahir plusieurs domaines d'applications différents, tels que le domaine militaire, scientifique, industriel, médical, climatique, etc. qui sont détaillées dans cette section.

1.5.1 Applications militaires

Les réseaux de capteurs sans fil sont appliqués avec beaucoup de succès dans la surveillance militaire. 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.[1]

Figure 1.10: Utilisation des RCSF dans le domaine militaire

1.5.2 Applications environnementales

Les réseaux de capteurs peuvent être utilisés pour surveiller les changements environnementaux [5]. Ils servent à déterminer les valeurs de certains paramètres à un endroit donné, comme par exemple : la température, la pression atmosphérique, humidité, de vibration, etc. En dispersant des nœuds capteurs dans la nature, on peut détecter des événements tels que des feux de forêts, des tempêtes ou des inondations. Ceci permet une intervention beaucoup plus rapide et efficace des secours. Avec les réseaux de capteurs on peut contrôler la pollution, par exemple en déposant des capteurs au-dessus d'un emplacement industriel pour détecter et surveiller des fuites de gaz ou de produits chimiques.

Figure 1.11: Utilisation des RCSF dans le domaine environnementale

1.6.3 Applications commerciales

Il est possible d'intégrer des capteurs au processus de stockage et de livraison dans le domaine commercial. Le réseau ainsi formé pourra être utilisé pour connaître la position, l'état et la direction d'un paquet. Il devient alors possible pour un client qui attend la réception d'un paquet, d'avoir un avis de livraison en temps réel et de connaître la localisation actuelle du paquet. 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 [1].

Figure 1.12: Les RCSFs pour les applications commerciales

1.5.4 Applications médicales

Les capteurs peuvent être implantés dans le corps humain pour contrôler les problèmes médicaux et pour aider les patients à maintenir leur santé en collectant des informations sur le patient : battement de cœur, tension du sang, la respiration. Ils peuvent être utilisés aussi pour surveiller les patients et l'avancement de leurs états dans un hôpital, mais de surveiller des malades ou des personnes âgées a distance. En outre, en implantant sous la peau des mini capteurs vidéo, on peut recevoir des images en temps réel d'une partie du corps sans aucune chirurgie et pendant environ 24h. On peut ainsi surveiller la progression d'une maladie ou la reconstruction d'un muscle.

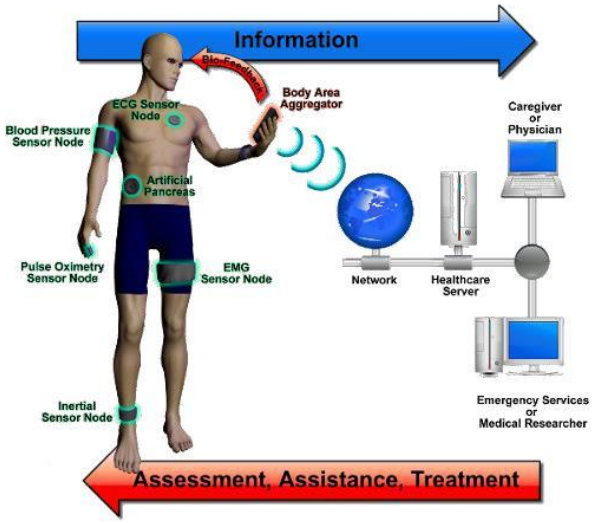

Figure 1.13: Utilisation des RCSF dans le domaine médicale

1.5.5 Applications domestiques

Les réseaux de capteurs peuvent également être utilisés dans la domotique et l'environnement intelligent [1]. Ils jouent un rôle essentiel dans les grandes usines et les entrepôts en surveillant les changements climatiques. Par exemple, des capteurs peuvent être utilisés pour contrôler les vibrations susceptibles d'endommager la structure d'un bâtiment. Dans les auteurs décrivent une application qui surveille l'état de grandes structures comme des immeubles administratif.

Figure 1.14: Utilisation des RCSF dans le domaine domestique

1.6 Classification des applications des RCSFS

Les applications des RCSFs peuvent ˆêtre classifiées en quatre classes distinctes : orientées temps (time-driven), orientées événements (event-driven), orientées requêtes (query-driven) et hybrides (hybride).

1.6.1 Applications orientées temps

Dans cette classe d'application, l'acquisition et la transmission de données sont liées au temps : instant précis ou période d'acquisition. Ainsi, la transmission de données dépend de la périodicité´ des mesures a` exécuter sur l'environnement local [7].

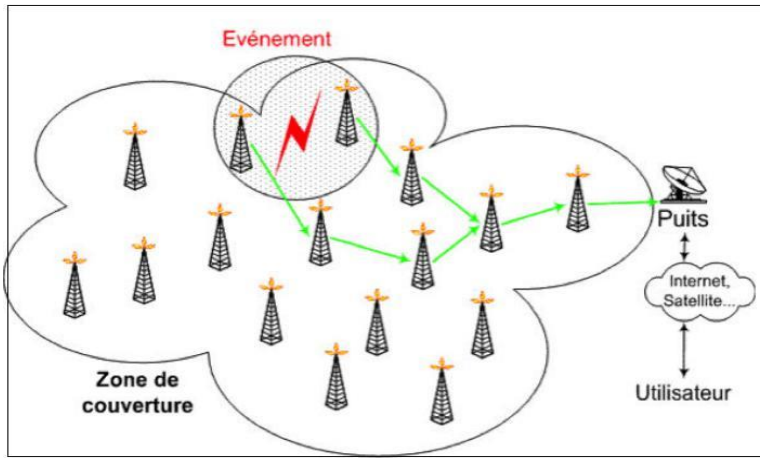

Figure 1.15: Collection des informations suite à un évènement

La collecte de données environnementales peut en représenter un bon exemple notamment dans le domaine de l'agriculture.

1.6.2 Applications orientées événements

Dans cette classe, les capteurs envoient leurs données seulement si un évènement spécifique se produit. On peut citer l'exemple de surveillance des feux dans les forêts. Au départ, cette classe d'application était conçue a` des fins militaires, comme la surveillance des déplacements des objets sur un champ de bataille. Par la suite, cette classe a rapidement trouvé de nouvelles perspectives dans les domaines[7].. : industriels, médical, sécuritaire ,etc.

1.6.3 Applications orientées requêtes

Dans cette classe, un capteur n'envoie sa donnée que lorsqu'une demande explicite de la station de base lui est parvenue. Ainsi, l'utilisateur peut interroger les capteurs pour acquérir des mesures

d'intérêts. Cependant, des connaissances sur la topologie du réseau et l'emplacement des capteurs peuvent s'avérer n´nécessaires pour l'interrogation d'un capteur bien déterminé´ [7].

1.6.4 Applications hybrides

Ce type d'application met en œuvre les trois modes de fonctionnement décrits précédemment. Par exemple, dans un réseau conçu pour le suivi d'objets, le réseau peut combiner entre un réseau de surveillance (time-driven) et un réseau de collecte de données par événements (event-driven). Ainsi, en plus d'un rapport périodique, lors de la détection d'un déplacement d'objet un message sera immédiatement transmis a` la station de base [7].

1.7 Caractérisation des réseaux de capteurs

La réalisation d'un réseau de capteur utilise les techniques des réseaux Ad hoc. Cependant, les protocoles et les algorithmes proposés dans ce dernier ne conviennent pas aux réseaux de capteurs sans fil. Voici donc quelques caractéristiques spécifiques à ces réseaux.

1.7.1 Topologie dynamique des réseaux de capteurs sans fil

La mobilité continue des noeuds crée un changement dynamique de topologie. Par exemple, un nœud peut joindre un réseau, changer de position ou quitter le réseau (suppression de capteurs à cause de défaillances ou autres choses, etc). Ce déplacement a naturellement un impact sur la morphologie du réseau [8].

1.7.2 Absence d'infrastructure

Les réseaux de capteurs en général, se distinguent des autres la propriété d'absence d'infrastructure préexistante et de tout genre d'administration centralisée à l'exception des réseaux ad hoc. Les hôtes mobiles sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue[8].

1.7.3 Un grand nombre de capteurs : des réseaux de 10000 nœuds peuvent être envisagés.

1.7.4 Sécurité physique limitée

Les réseaux de capteurs sans fil mobiles sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé [8].

1.7.5 Auto déploiement

Les noeuds capteurs peuvent être déployés dans l'environnement sans intervention humaine et demeurèrent sans surveillance pendant longtemps après déploiement [8].

1.7.6 Auto configuration

Consiste avant tout à permettre la communication des noeuds, et donc l'affectation d'adresses aux interfaces réseaux, la diffusion des préfixes des sous réseaux et des passerelles d'interconnexion. Il s'agit ensuite de diffuser les informations comme le DNS aux noeuds du réseau[8]

1.7.7Auto-organisation

Cherche à structurer la topologie du réseau en tirant partie des propriétés des noeuds tels que l'énergie résiduelle, la densité, etc. Par exemple, est-il préférable de proposer une architecture à plat du réseau ou au contraire est-il nécessaire de proposer des mécanismes regroupant les noeuds suivant des critères comme les services, les interfaces radio, les capacités, l'énergie disponible, etc. L'auto-organisation doit également répondre à la prise en compte efficace de la dynamicité du réseau [8].

1.7.8 Auto gestion

A pour objectif de proposer une supervision autonome : une fois un réseau spontané auto-configuré et auto-organisé, il est nécessaire de fournir des algorithmes et des protocoles permettant sa surveillance, sa maintenance réactive [8].

1.7.9 Contrainte d'énergie

Dans plusieurs applications, les noeuds de capteurs sont placésdans des surfaces distantes, le service du noeud peut ne pas être possible, dans ce cas la durée de vie du nœud peut être déterminée par la vie de la batterie, ce qui exige la minimisation des dépenses énergétiques [8].

1.8 Conclusion

Dans ce chapitre nous avons présenté les réseaux de capteurs sans fil (RCSF), en exposant leurs architectures, leurs contraintes ainsi que leurs domaines d'applications.

Dans le chapitre suivant, nous allons présenter le principe de localisation utilisé dans des systèmes tels que GPS et Galileo, puis nous abordons la problématique de la localisation dans les réseaux de capteurs sans fil (RCSF).

Chapitre 2

Localisation dans les RCSFS

Sommaire

2.1 Introduction

La capacité de localisation (estimation de la position) est essentielle dans la plupart des applications de RCSF. Les procèdes employés pour déterminer une position sont basés sur des calculs géométriques comme la triangulation et la trilatération.

Pour connaitre la distance entre deux nœuds, plusieurs techniques peuvent être utilisées, comme la synchronisation, la puissance de signal reçu ainsi que les caractéristiques physiques de l'onde porteuse. D'autres approches, comme les caractéristiques du signal radio reçu et l'angle de l'arrivée peuvent être également appliquées pour le calcul de position. Les techniques de localisation dans les RCSFs sont utilisées pour estimer l'emplacement des capteurs sans position connu auparavant dans le réseau en utilisant les informations de position de quelques capteurs [9].

Dans ce chapitre, nous allons définir la localisation, ses objectifs, ses propriétés, ses systèmes et ses critères, ensuite nous introduisons un échantillon représentatif des différents protocoles de mesure et algorithmes de calcul de position disponibles. Enfin nous terminons avec un tableau comparatif et une synthèse.

2.2 Localisation

2.2.1 Définition

La localisation est un procédé´ permettant de positionner un objet sur un plan ou une carte géographique, cette opération est réalisée a` l'aide d'un terminal capable d'être localisé en temps réel. La localisation dans les RCSFs est un problème qui a reçu beaucoup d'attention en raison de l'importance d'obtenir des informations de position fiable[10].

2.2.2 Objectifs de la localisation

- Déterminer les coordonnées géographiques des différents capteurs.
- Pour le développement de protocoles de routage de l'information récoltée.
- Pour la couverture de la zone d'intérêt.
- Pour l'estimation des distances entre les nœuds [10].

.

2.2.3 Propriétés de localisation

1- Un système de localisation doit avoir les propriétés suivantes :

2- Une technique d'estimation de position (Trilatération ou triangulation).

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

- Les positions absolues renseignent sur la position réelle de l'objet sur le globe terrestre (longitude et latitude) ou dans l'espace (longitude, latitude et altitude).
- Les positions relatives indiquent juste une direction par rapport `a un voisinage donné à droite au bout de la rue par exemple.
- Les positions symboliques désignent par exemple une salle, un espace particulier.

3 - Une précision de position :une position peut aller d'un point dans le cas d'une grande précision `a une surface (ou volume) si la précision de position est moins importante.

4- Une architecture particulière : un système de positionnement en intérieur dans un bâtiment par exemple ne possède pas les mêmes contraintes qu'un système de localisation d'extérieur.

5- Un coût (matériel, infrastructure,. . .)[11].

2.2.4 Principaux systèmes de localisation

- **GPS : (Global Positioning System)**

Est le système de localisation américain. Opérationnel depuis les années 1980, il a été développé pour fournir à l'armée américaine un système de repérage à couverture mondiale et de très grande précision. Son rôle consiste, par exemple, à guider un missile sur des centaines de kilomètres. Son nom officiel est NAVSTAR system (Navigation Satellite Timing and Ranging).

C'est à la fin de l'année 1993 que le département américain de la défense a ouvert l'accès gratuit au GPS pour les utilisateurs civils [CS09].

Les performances du système de localisation sont directement liées aux performances de chaque partie. La Figure (2.1) illustre la décomposition du système de localisation en sous systèmes. Les nœuds représentés par des cercles gris sont des ancres, les autres nœuds (en blanc) sont des nœuds qui ne sont pas encore localisés.

- **Système russe GLONASS (IAC)**

Développé de 1976 à 1982, n'est plus pleinement opérationnel et ce dû aux conditions politiques et économiques du pays. Toutefois, la Russie a entrepris sa remise à niveau et GLONASS devrait être de nouveau fonctionnel en 2011.

- **Système IRNSS (Indian Regional Navigational Satellite System)**

De son côté, l'Inde met en œuvre son système IRNSS. Il offrira une précision au sol inférieur à 20 mètres et devrait être prêt en 2012.

- **Système Beidou**

Développé par la Chine en est à une version expérimentale (Beidou-1). Elle est composée de quatre satellites ayant des fonctionnalités limitées. La version Compass (ou Beidou-2) devra compter 35 satellites opérationnels d'ici :

- **Système Galileo**

Enfin, Galileo (CNES et ESA) est le système de localisation européen. Il est composé de 27 satellites et atteindra une précision inférieure au mètre pour les applications du domaine civil [CS09]:

2.2.5 Importance des systèmes de localisation

Les positions des capteurs peuvent être prédéterminées et préconfigurées. Mais, en général, les capteurs ont un déploiement aléatoire, car ils sont utilisés sur des terrains inaccessibles, embarqués sur des engins mobiles où sur le lieu d'un désastre ou leur nombre (très grand) ne permet pas de préconfigurer la position.

Pour cela, un système de localisation est nécessaire afin de fournir aux nœuds leurs positions [12].

- **Identification des données collectées**

Cela consiste à projeter les données/événements sur la position de collecte/occurrence. L'un des objectifs majeurs des RCSF est de sur-veiller une zone d'intérêt. Cependant, après que les données soient collectées, il est important d'identifier la région d'où viennent ces données[12].

- **Agrégation des données collectées**

Cela permet aux nœuds intermédiaires de corréler et de fusionner les données qui proviennent de la même région quand ces données sont transmises via le réseau[12].

- **Adressage des nœuds**

Cela permet d'utiliser la position des nœuds comme identifiant unique dans le réseau.

- **Administration du réseau**

Cela permet d'administrer et d'interroger les nœuds localisés dans une région, évaluer la couverture des nœuds, et de générer une carte d'énergie disponible dans chaque nœud[12].

- **Algorithmes géographiques**

Ce sont des algorithmes qui utilisent l'information de localisation des nœuds pour optimiser l'utilisation des ressources du RCSF. Parmi ces[12].

2.2.6 Composition d'un système de localisation

Un système de localisation peut être décomposé en trois parties distinctes. Chaque partie a son propre objectif et méthodes de résolution. Elles seront étudiées séparément. Ces trois parties (sous-systèmes) sont :[10].

- **Estimation de distance/angle**

Cette partie permet d'estimer la distance et/ou l'angle entre deux nœuds. Cette information est utilisée par la suite par les deux autres parties (estimation locale).

- **Calcul de la position**

Cette partie permet d'estimer la position d'un nœud en se basant sur les mesures de distances et/ou d'angles disponibles et sur les positions des nœuds de références (ancres) en utilisant la triangulation par exemple.

- **Algorithme de localisation**

C'est la partie la plus importante du système de localisation. Elle définit la manière avec laquelle les informations.

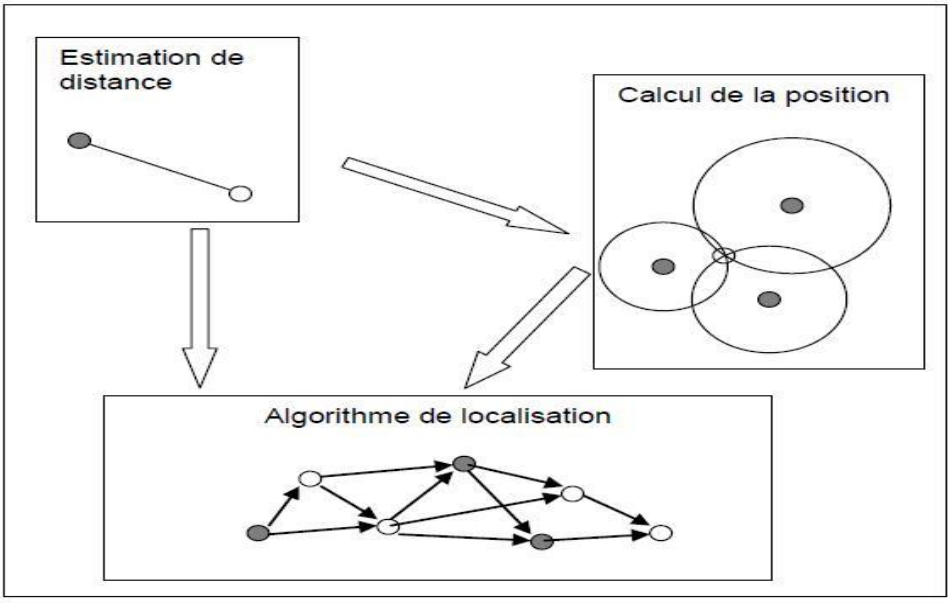

Figure 2.1: Composition d'un système de localisation.

2.3 Algorithmes de localisation dans les réseaux de capteurs

L'algorithme de localisation est le composant principal d'un système de localisation. Cette composante détermine comment l'information des distances et des positions sera manipulé afin que la grande partie, ou la totalité des nœuds du RCSFs estiment leurs positions[12].

2.3.1 Catégories des algorithmes de localisation

- **Distribuée ou centralisée**

Les positions des nœuds peut être calculé d'une manière distribuée par chaque nœud de réseau, ou d'une manière centralisée par un seul nœud central[12].

- **Avec ou sans infrastructure**

S'il n'y a pas besoin d'infrastructure ou s'il est nécessaire de repenser l'infrastructure précédente, afin de permettre le fonctionnement de l'algorithme de localisation.

- **Positionnement relatif ou absolu**

Les positions calculées peuvent être liées à coordonnées globales (par exemple, latitude, longitude) ou liées à un noeud ou un point du réseau[12] .

- **Scénarios intérieure ou extérieure**

Si le système est plus approprié aux applications à l'intérieur (indoor) ou plutôt à l'extérieur (outdoor) [12].

- **A un saut ou multi-saut**

Si tous les nœuds inconnus ont une communication directe avec les noeuds de balise(ancres) ou si une communication multi-sauts est nécessaire

2.3.2 Paramètres de performance d'un algorithme de localisation

Plusieurs paramètre peuvent être utilisés pour évaluer les performances d'un algorithme de localisation :[12]

- **Erreur moyenne et consistance**

Cette identifie est l'erreur moyenne de l'estimation distance et montre également si cette erreur moyenne est répété dans des scénarios similaires, mais déférentes (de consistance de la moyenne). Cette erreur moyenne limite l'utilisation de système de localisation pour les applications où ce niveau d'inexactitudes est acceptable.

- **Coût de communication**

Cela permet de déterminer la complexité de l'algorithme de localisation en termes de paquet échangés. Il détermine aussi le coût du système de localisation de nœud du réseau.

- **Nombre de nœuds localisés (fixés)**

Ceci définit le pourcentage de nœuds de réseau qui ont reussi à calculer leur position à la fin de l'algorithme de localisation. L'idéal est que tous les nœuds doivent être capables de calculer leur position, mais dans de nombreux cas, il n'est pas possible.

- **Nombre des nœuds de balise(les ancres)**

Définit le nombre des nœuds de balise nécessaires pour que l'algorithme de localisation marche. En général, les Nœuds de balises sont plus chers que le nœuds normales, et leur utilisation doivent être réduits au minimum.

2.3.3 Caractéristiques de la performance d'un système de localisation dans les RCSFS

Plusieurs caractéristiques affectent les performances d'un système de localisation. Pour chaque système de localisation, il est important de faire plusieurs expérimentations (obtenir des données statistiques) afin d'évaluer son comportement lorsque les paramètres suivant changent [12].

- **Densité du réseau**

Elle détermine le nombre de nœuds par unité de surface. Dans réseaux à forte densité, les distances entre les nœuds sont courtes ce qui implique des erreurs lors de l'estimation des distances, et par conséquent sur les erreurs du système de localisation. De plus, le grand nombre de nœuds non-localisés ce que leur permet de mieux estimer leurs positions.

- **Taille du réseau**

l'augmentation du nombre de nœuds (et en gardant la même densité ce qui revient à élargir la surface du réseau), se traduit par un nombre plus élevé de sauts. En général, l'augmentation du nombre de sauts s'accompagne par l'augmentation de l'imprécision[13]due à l'accumulation des erreurs.

- **Nombre des nœuds de balise(les ancres)**

Lors du déploiement d'un plus grand nombre de nœuds de balise dans le réseau, l'erreur moyenne du système de localisation a tend à diminuer et le nombre de nœuds installés tend à augmenter.

- **Précision GPS**

plusieurs travaux utilisent le GPS pour localiser les noeuds de balise(les ancres).le système GPS n'est pas parfait et sa précision va influer les performances du système de localisation.

Dans les sections suivantes, des algorithmes de localisation proposées seront étudiées et analyser. Ces algorithmes sont : les Algorithmes DV-hop, algorithme faible complexité et énergie efficace, Algorithme hybride CO-WVSS[13].

2.4 Algorithmes DV-hop

2.4.1 DV-hop traditionnel

Dans algorithme DV-hop [14], chaque nœud d'ancrage calcule sa distance moyenne d'un saut et se propage vers les nœuds de capteurs.

$$HopSize_i = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{i \neq j} h_{ij}} \quad (1)$$

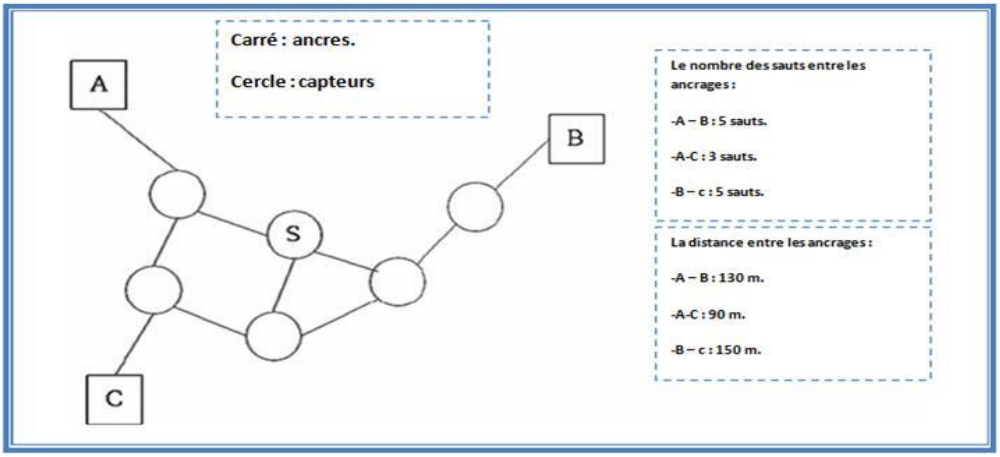

Figure 2.2: Exemple d'algorithme DV-Hop

Tableau 2.1: Les distances moyennes et l'estimation de distance en utilisant la méthode DV-Hop

Soit 130 la distance entre A et B ,90 la distance entre A et C ,150 la distance entre B et C

Soit 5 le nombre des sauts entre le A et B, 3 le nombre des sauts entre A et C, 5 le nombre des saut entre B et C.

	A	B	C
la distance moyenne d'un saut	$\frac{130+90}{5+3}$	$\frac{130+150}{5+5}$	$\frac{150+90}{5+3}$
la distance à partir d'un nœud d'ancre	$\frac{130+90}{5+3}$	$\frac{130+150}{5+5}$	$\frac{150+90}{5+3}$

(x_i, y_i) , (x_j, y_j)les coordonnées de nœud i et le nœud j respectivement.

h_{ij} le nombre minimum des sauts entre le nœud j et i.

$$d_{ij} = hopSize_i \times hopCont_{ij} \quad (2)$$

d_{ij}: La distance de nœud j vers le nœud i.

$HopCount_{ij}$: le nombre minimum des sauts de nœud j vers le nœud i.

Chaque nœud de capteur estime la distance entre les nœuds d'ancrages et lui-même en fonction

De la distance moyen d'un saut et le nombre des sauts. Dans l'exemple de la figure(2.2), ils existes trois nœud d'ancrages(A,B,C) et un nœud capteur S pour estimer sa distance. A est le nœud d'ancrage le plus proche du nœud capteur S.

La distance moyenne d'un saut de A est utilisée pour l'estimation de la distanc

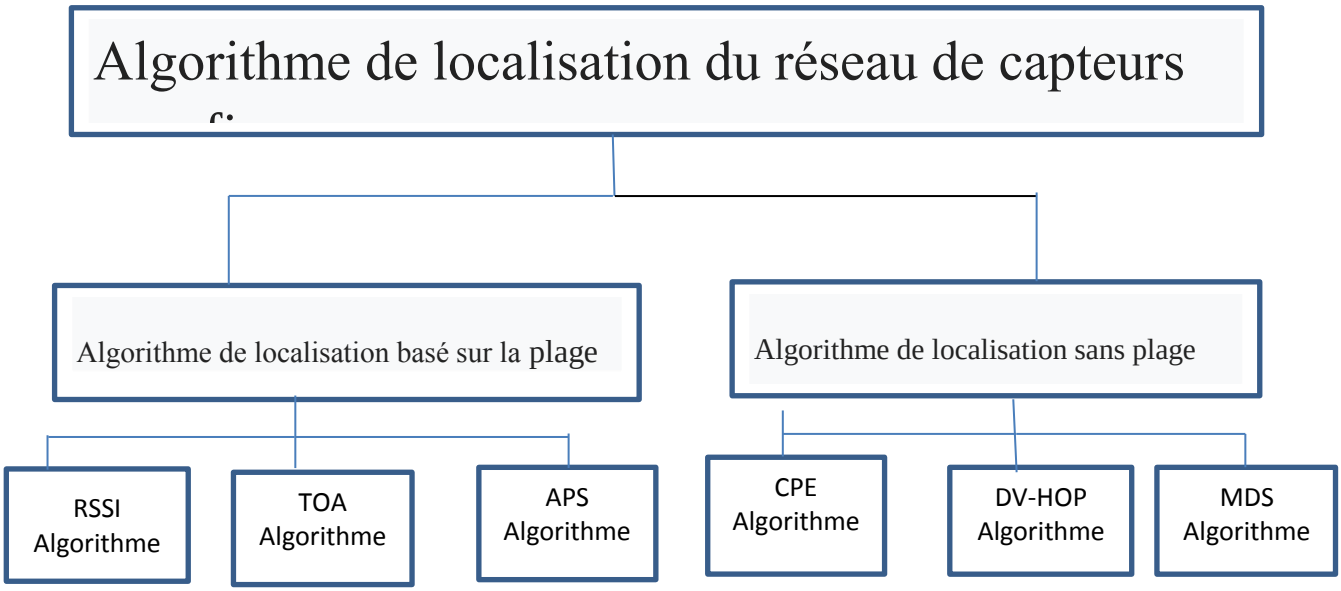

Figure 2.3: Classification diagramme de algorithme localisation

2.4.2 FDV-Hop

Un algorithme de localisation FDV-Hop [15] a été proposé pour déterminer les informations de position des nœuds inconnues. FDV-Hop utilise l'idée de pondération et de validation pour améliorer DV-Hop.

Description d'algorithme

FDV-Hop commence par la diffusion des messages de balise de tous les nœuds ancres via leurs emplacements aux autres nœuds inconnu , les nœuds ancres calculent la distance moyenne d'un saut.

$$s_i = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{i \neq j} h_{ij}} \quad (3)$$

(x_i, y_i) Les coordonnées de nœud ancre i.

h_{ij} le nombre minimun des sauts entre le nœud i et j.

Lorsque le nombre d'ancrages dans le réseau est M, l'ancre calcule l'erreur de sa taille moyenne d'un sauts par :

$$\varepsilon_i = \frac{\sum_{I \neq J}^{M} |d_{rij} - d_{eij}| / h_{ij}}{M-1} \quad (4)$$

la distance réelle entre l'ancre i et j est représentée par d_{rij}, calculé par :

$$d_{rig} = \sqrt{(x_i - x_j)^2 + (y_i + y_j)^2} \quad (5)$$

La distance de mesure entre l'ancre i et j, qui est représenté par d_{eij} est calculé par

$$d_{eij} = s_i + h_{ij} \quad (6)$$

La valeur pondérée de chaque ancre est

$$w_i = \frac{\frac{1}{\varepsilon_i}\frac{1}{N_i}}{\sum_{j=1}^{M}(\frac{1}{\varepsilon_j}\frac{1}{N_i})} \quad (7)$$

N_i Désigne le nombre de sauts entre le nœud inconnu et l'ancre i.

Puis le nœud utilise La valeur pondérée et la taille moyenne d'un saut des d'ancre pour calculer sa propre distance moyenne d'un saut.

$$s = \sum_{i=I}^{M} w_i s_i \quad (8)$$

Un nœud inconnue a sélectionné les trois nœuds d'ancrages (L_1, L_2, L_3) les plus proches pour calculer les distances (d_1, d_2, d_3), (x,y) les coordonnées nœud inconnue et (x_i, y_i) nœud d'ancre ,la formule de la méthode de triangulation sur la base de validation est

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = (s \text{ x} h_1)^2 \\ (x_2 - x)^2 + (y_2 - y)^2 = (s \text{ x } h_2)^2 \\ (x_3 - x)^2 + (y_3 - y)^2 = (s \text{ x} h_3)^2 \\ \sqrt{(x_i - x)^2 + (y_i - y)^2} \leq R \ldots\ldots h_i l, i = 1,2,3 \\ (h_i - 1)R \leq \sqrt{(x - x)^2 + (y - y)^2} \leq h_i \text{x} R \ldots\ldots\ldots h_i \neq 1, i = 1,2,3 \end{cases} \quad (9)$$

R Est le rayon moyen de communication de réseau et h_i représente le nombre de sauts de l'ancre i au nœud inconnue.

Dans cet algorithme, basé sur la méthode de triangulation, la coordonnée du nœud inconnue est calculée dans un premier temps. Ensuite, la coordonnée obtenue est validé par les deux dernières formules de validation.

Si le nombre de sauts est inférieur à 1, alors la distance réelle de deux nœuds doit être inférieur au rayon de la communication. Alors que si le nombre de sauts est supérieur à 1, alors la distance réelle

de deux nœuds doit être comprise entre le nombre de sauts et le rayon de la communication. Selon l'algorithme ci-dessus, Si les résultats ne peuvent pas satisfaire les résultats de la validation, ce que nous devons faire est de répéter les étapes ci-dessus jusqu'à ce que les coordonnée des nœud ancre soit calculer.

2.4.3 DV-Hop amélioré

La plupart des algorithmes améliorés DV-Hop se concentrent sur la précision des algorithmes mais sans améliorer la consommation d'énergie, dans l'article [16] ils ont proposé une nouvelle amélioration de la méthode DV-Hop qui augmente le nombre des nœuds d'ancrages par la mise en niveau des nœuds positionnés pour qu'ils être des nœuds nouveau-né ,ces nœuds permet de crée un système d'ancrage économique pour accélérer la localisation des nœuds inconnus et réduire l'ensemble des tours localisation, ainsi favorise l'économie d'énergie

Description d'algorithme

Dans cette algorithme une centaine de nœud sont déployés de manière aléatoire dans une région, les ancres nécessaires pour localiser les nœuds inconnus d'un réseau reçoit l'information à partir SB (station de base), L'algorithme commence par la diffusion de tous les nœuds ancres via leurs emplacements aux autres nœuds[16] .

Quand le nœud inconnu reçoit l'information, il calcule la distance moyenne d'un saut qui est basé sur deux facteurs

- La distance moyenne d'un saut de nœud d'ancrage
- La distance moyenne pondérée d'un saut de l'ancre

$$S = \sum_{i=1}^{N} W_i * S_i \quad (10)$$

S_i : C'est la distance moyen d'un saut de nœud d'ancrage,

W_i : La distance moyenne pondérée d'un saut de l'ancre i est donnée par :

$$W_i = \frac{(h_i+G_i)^{-1}}{\sum_{j=1}^{N}(h_i+G_j)^{-1}} \quad (11)$$

N : le nombre des nœuds, G_i : la génération de l'ancre i, h_i

: le nombre de sauts entre nœud inconnue et l'ancre i,

Après la distance moyenne d'un saut, le nœud inconnue peut calculé sa distance a l'aide de la trilateration ou multilatération.

Les règles de localisation de l'algorithme :

Afin de réduire les frais généraux de communication et d'éviter l'utilisation de l'information redondante, les règles de localisation de l'algorithme sont les suivantes :

- Dans le premier tour de la localisation, si les nœuds inconnues reçoivent la stern(paquets de données) des ancres qui pourraient être combinés de manière efficace, puis ces nœuds se localiser et la mise à niveau pour être des ancres nouveau-nés, et ils diffuse ensuite les paquets de données reçus, y compris l'emplacement et la génération d'informations.

- dans le deuxième tour, quand les nœuds inconnues reçoivent la sterne d'ancre qui ne sont pas combinaisons d'environnement colinéaires, se localisant selon la méthode suivante :

Si les nœuds inconnus reçoivent la sterne seulement constitué de deux ancres de départ et un point d'ancrage du nouveau-né, ils se localiser directement.

Si les nœuds inconnus reçoivent plusieurs ancres, ils choisissant quatre ancres avec moins de sauts a partir des nœuds d'ancrages et moins générations, Pui ils se localiser. Les nouveaux noeuds situés sont mis à niveau que de nouvelles ancres et propagent l'information lorsque la génération est inférieure au seuil[16].

Si les nœuds ne concordent pas avec les conditions ci-dessus, ils attendent pour recevoir de nouvelles informations provenant d'autres ancres.

3. La localisation subséquente est la même que le deuxième tour. Le processus de localisation est considéré comme fin jusqu'à ce que le tour de certains nœuds positionnement inconnu vient aux limites : $\max_{\text{tour}}$

2.4.4 GGDI

Dans [17] les auteurs décrivent un algorithme de localisation à économie d'énergie et faible complexité. ils ont utilisé un nœud d'ancrage mobile dispose d'un récepteur GPS pour définir ses coordonnées de localisation en temps réel et à énergie suffisante pour diffuser des messages pendants processus de localisation et d'antenne à faisceau unique, il se déplace en ligne droite dans la zone du capteur et chaque nœud capteur est équipé d'un antenne omnidirectionnel. Le nœud mobile n'a pas d'énergie limité, et chaque nœud capteur fixe à une énergie limitée. Description d'algorithme

Les nœuds de capteurs sont déployer aléatoirement dans la zone de capteur, et chaque capteur reçoit un maximum RSSI sur plus grands ligne de direction de gain d'intersection(GGDI).chaque nœud de capteur trouve deux grand lignes dans la zone de couverture d'un nœud mobile d'ancrage (figue 3-2).le point commun des deux lignes c'est l'emplacement de nœud capteur. En GGDI nous utilisons les points de balise pour obtenir l'emplacement. - N_1 c'est la position de

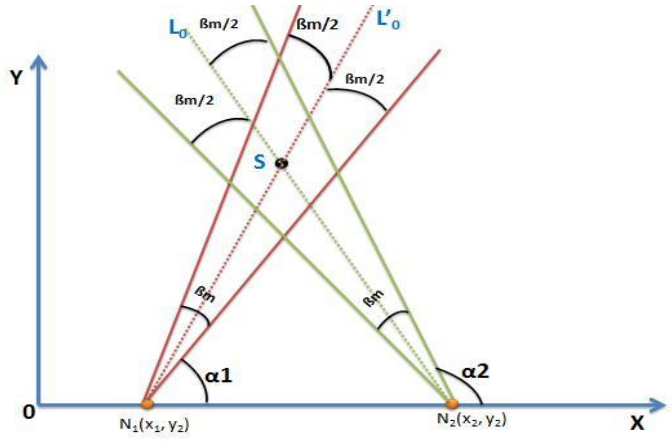

Figure 2.4: Localisation d'un nœud inconnu avec la méthode GGDI

Nœud de capteur pour recevoir un paquet avec un maximum RSSI à partir de nœud d'ancrage Mobile pour la première fois[17].

- N_2 c'est la position de noeud capteur pour recevoir un paquet avec un maximum RSSI pour la deuxième fois.

Chaque noeud de capteur maintient une liste de visite qui contient des valeurs RSSI visité et des informations sur les paquets. La liste d'emplacement comprend points de balise N_1, N_2 et $\alpha, \alpha 2, \beta_m$ m est déterminer sur la base de la liste des visiteurs.

Après que chaque noeud de capteur obtient deux points de balise, on peut calculer l'emplacement de noeud capteur :

N_1, N_2 Sont des balises dont les coordonnées sont :

(x_1, y_1) et (x_2, y_2) respectivement.

- L'angle entre L_0 et l'axe x est $\theta = \alpha 2 + \beta_m/2$
- l'angle entre $L_0^{'}$ et l'axe x est $\varphi = \alpha 1 + \beta_m/2$

Les coordonnées de chaque capteur peuvent être obtenues par les équations suivantes :

$$\begin{cases} L_0 : y - y_{1=}\ k_1(x - x_1) \\ L_0^{'} : y - y_{1=}\ k_1(x - x_1) \end{cases} \quad (12)$$

Le point d'intersection de deux lignes Ł0, L'0 est l'emplacement du capteur dont les coordonnées sont :

$x = (\mathrm{a} - \mathrm{b})/(k_2 - k_1$; $y = (\mathrm{a} \times k_2 - \mathrm{b} \times k_1)/(k_2 - k_1)$ Avec :

$a = y_1 - k_1 \times x_1$

$b = y_2 - k_2 \times x_2$

$k_1 = \tan(\theta)\ et\ k_2 = \tan(\varphi)$

2.4.5 CO-WVSS

Dans [18] , un algorithme hybride de localisation pour améliorer la précision en fonction de RSSI / AOA est proposé. Cependant, en raison du non linéaire de la fonction de coût basée sur RSSI, les chercheurs ont proposé une méthode numérique pour corriger et optimiser la valeur de poids à partir de la puissance du signal de similitude appelé CO-WVSS, cette méthode visent
à deux processus de l'algorithme de localisation pour améliorer la précision et combinée avec algorithme de localisation triangle barycentre simple basé sur RSSI. Puis le rangement radial avec AOA.

L'idée fondamentale de triangle barycentre est : La relation entre trois nœuds de balises (A, B, C) qui sont le centre d'un cercle et un nœud inconnu(M) est un triangle tel que D, F, G est les sommets[19], le nœud inconnu M est le centroïde de triangle, comme cela est représenté sur

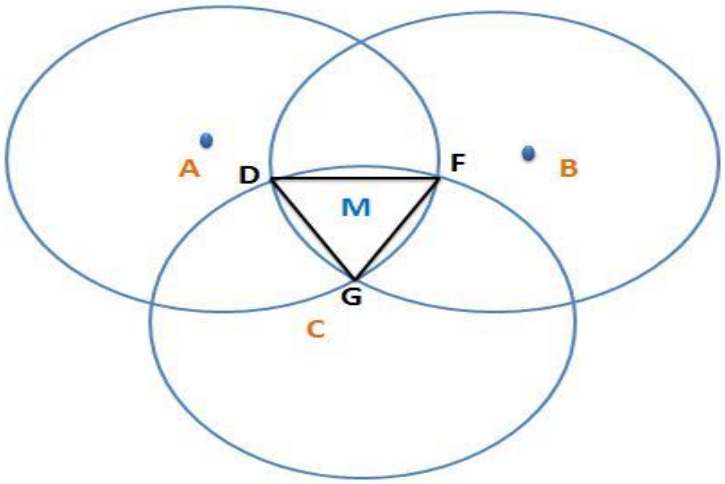

Figure 2.5: Algorithme de localisation Triangle Centroid

.

r_a, r_b, r_c ont la distance avec le nœud inconnu respectivement. Les coordonnées de point D peut obtenue par l'équation (13).

$$\begin{cases} (x_M - x_a)^2 - (y_M - y_a)^2 \leq {r_a}^2 \\ (x_M - x_b)^2 - (y_M - y_b)^2 \leq {r_b}^2 \\ (x_M - x_c)^2 - (y_M - y_c)^2 \leq {r_c}^2 \end{cases} \quad (13)$$

De même, les coordonnées des points F et G peuvent été trouvé.

Ensuite, est supposé que les coordonnées du nœud inconnu est $M(x_{m,}x_m)$, les coordonnées du nœud inconnue M peut être obtenue par l'algorithme de centroïde [20]. La coordonnée M est représentée dans l'équation (14)

$$\begin{cases} x_m = \frac{x_e+x_f+x_g}{3} \\ y_{m=} \frac{y_e+y_f+y_g}{3} \end{cases} \quad (14)$$

Description d'algorithme

Algorithme de localisation hybride CO-WVSS se compose de trois phases principales :

- **Première phase**

Recenser l'erreur de mesure de RSSI par la contrainte géométrique de distance, puis le nœud inconnu par algorithme de localisation triangle centroide. La coordonnée du nœud M est montrée dans l'équation (14).

- **Deuxième phase**

Estimer l'angle et le chemin par AOA, placer M avec aWv (15) (la valeur de poids) sur le chemin radial au maximum RSSI de la mesure de trois nœuds d'ancre comparativement, AOA est montré dans l'équation (16).

$$aWv = \lg(P_2/P_1)\lg(d_1/d_2) \quad (15)$$

Avec P1 et P2 sont des puissances moyenne des deux noeuds, et par exemple lg P1 = RSS1 /10

$$\sin\theta/\cos\theta = \left(x_f - x_m\right) / (y_f - y_m) \quad (16)$$

- **Troisième phase**

Le point F est obtenue par l'algorithme de centre de gravité de triangle (triangle centroide) dans un environnement idéal, est l'ange radiale, MF est le chemin radiale, comme le montre la figure (2.5).

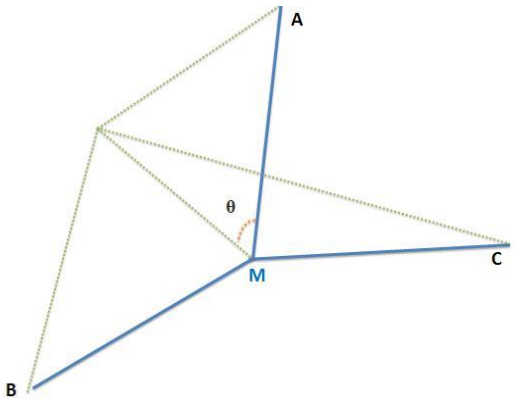

Figure 2.6: Corriger et optimiser la localisation

2.5 Comparaisons des algorithmes de localisation

L'algorithme de localisation est le composant principal d'un système de localisation. ce composant définit la façon dont l'information disponible est fournie par les nœuds d'ancrages, par les méthodes d'estimations de distances, et par les combinaisons de distance seront manipulés [12]. le tableau suivant décrit une comparaison entre les algorithmes présentés dans ce chapitre.

Tableau 2.2: Comparaison les algorithmes de la localisation

Algorithme	Nombres des ancres	calcul de la position	infrastructure	Scénarios	multi-sauts
FDV-Hop	> 3	Distribué	Non	En plein air	Oui
DV-Hop améliorée	> 3	Distribué	Non	En plein air	Oui
GGDI	1 mobile	Distribué	Oui	En plein air	Non
CO-WVSS	3	Distribué	Non	En plein air	Non

.

2.6 Conclusion

Dans ce chapitre nous avons présenté un état de l'art sur les algorithmes de localisation dans les réseaux de capteurs sans fils. L'étude et l'analyse de ces principaux algorithmes et approches de localisation nous a permis de proposer notre propre algorithme de localisation dont l'objectif principal est le prolongement du temps du vie du réseau ainsi que la gestion efficace de la consommation énergétique. Les performance de notre algorithme de localisation proposé vont être présentés dans le chapitre qui suit.

Chapitre 3

Algorithme des chauves-souris

Sommaire

3.1 Introduction

L'algorithme des chauves-souris, dont l'appellation d'origine est Bat Algorithme, est une métaheuristique très récente. Le premier article la concernant a été proposé en 2010 par Xin-She Yang [21]. Elle est basée sur le comportement d'écholocation des chauves-souris. Spécifiquement, l'espèce des microchiroptères, qui se dirigent en vol par écholocation, peuvent trouver, discriminer les différents types d'insectes et éviter les obstacles, cela même dans l'obscurité totale [22].
Dans ce chapitre nous allons en premier lieu aborder brièvement le comportement d'écholocation des chauves-souris, dans un second sa formulation.

3.2 Heuristiques ou méta-heuristiques

Une méthode heuristique s'arrête lorsqu'elle arrive à construire une solution pour le problème, par contre une méta-heuristique s'arrête lorsqu'un critère d'arrêt est atteint [23].
La plupart des heuristiques et des méta-heuristiques utilisent des processus aléatoires et itératifs pour récolter de l'information, explorer l'espace de recherche et faire face à des problèmes comme l'explosion combinatoire.
L'espace de recherche d'une méta-heuristique représente l'ensemble des solutions du problème qu'elles soient bonnes ou mauvaises, alors qu'une approche heuristique explore l'espace des états du problème en vue de construire la solution [23].
Les méta-heuristiques sont classées en deux catégories

- **Les méthodes à population de solutions** connues sous le nom d'algorithmes évolutionnaires comme les algorithmes génétiques…
- **Les méthodes à solution unique** comme le recuit simulé.

3.2.1 Principe de voisinage

À chaque solution **s** d'un problème, on associe un sous-ensemble V(S) de solutions. Une méthode de voisinage débute généralement avec une configuration initiale **s** à laquelle un processus itératif est appliqué. Il cherche à améliorer la configuration courante en la remplaçant par une de ses voisines en tenant compte de la fonction objective. Ce processus s'arrête et retourne à la meilleure solution trouvée lorsque le critère d'arrêt est atteint. Cette condition d'arrêt concerne généralement une limite sur le nombre d'itérations ou sur l'objectif à réaliser [24].

La figure ci-dessous présente les différentes méthodes d'optimisation méta-heuristiques.

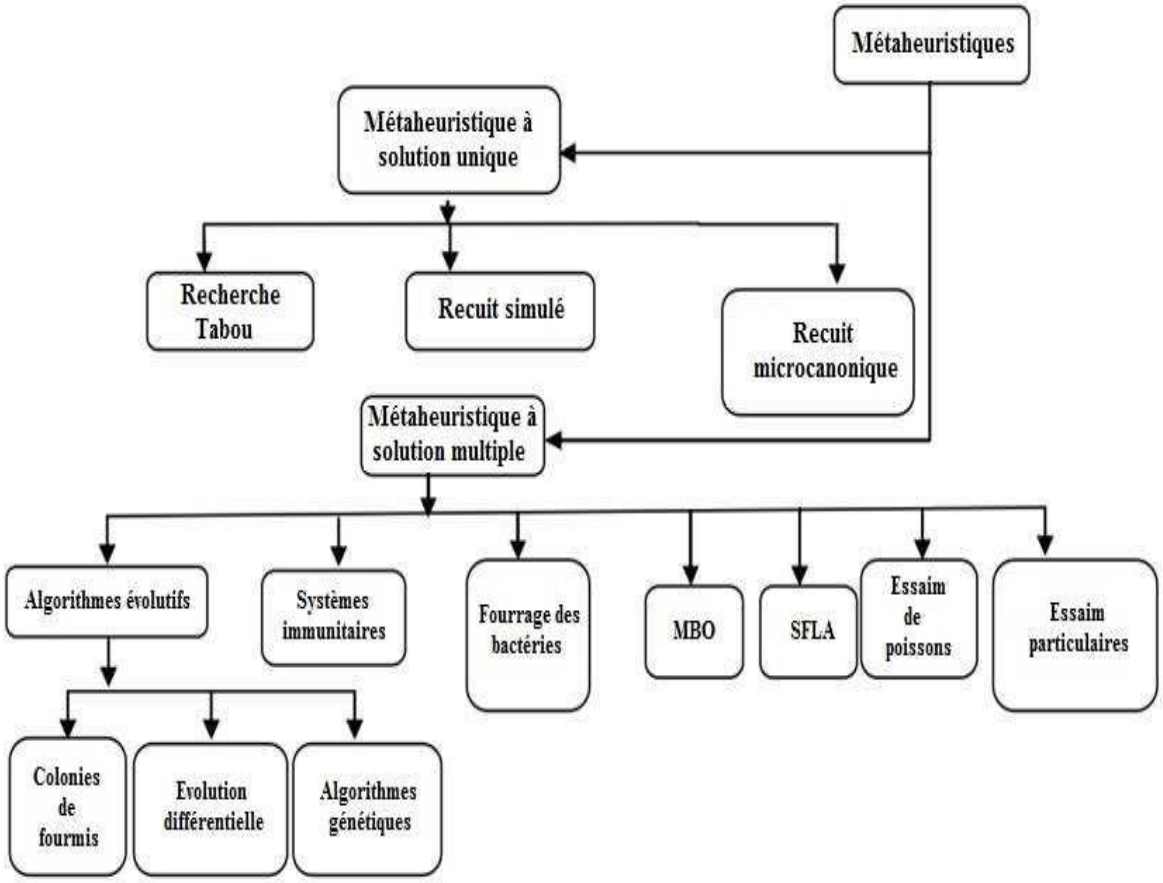

Figure 3.1: Méthodes d'optimisation méta-heuristiques [25]

3.3 Algorithme de chauves-souris (BA)

3.3.1 Définition

L'algorithme des chauves-souris est un algorithme d'optimisation méta-heuristique développé par Xin-She Yang en 2010.

3.3.2 Echolocation des chauves-souris

Les chauves-souris sont les seuls mammifères avec des ailes. Il existe 2 types de chauves-souris: les Mégachiroptères (méga-chauves-souris) et les Microchiroptères (micro-chauves-souris) [26].

3.3.3 Comportement des chauves-souris

Les chauves-souris qui utilisent l'écholocation ont la capacité de déterminer l'environnement autour d'eux, ils peuvent détecter la distance et l'orientation de la cible (proie) ainsi que l'emplacement des obstacles.

La nature et le rythme, le temps de retard de la réponse, le volume et la différence de temps entre les deux oreilles de la chauve-souris lui permettent de dessiner l'image 3D de l'environnement dans son cerveau.

Les chauves-souris volent au hasard avec une vitesse vi à la position xi avec une fréquence F_{min} fixe, et font varier la longueur d'onde ʎ, la fréquence F et le volume A_0 lors de la recherche d'une proie.

- Si aucune proie n'est dans la portée de la chauve-souris, la valeur de la longueur d'onde est augmentée, et la fréquence est diminuée.
- Si la proie est proche de la chauve-souris, la longueur d'onde est réduite, et la fréquence est augmentée pour mesurer précisément l'emplacement de la proie et la suivre plus rapidement [26].

Les chauves-souris peuvent automatiquement régler la longueur d'onde (ou fréquence) de leurs impulsions émises et ajuster le taux d'émission de l'impulsion **r** ϵ [0, 1] qui varie en fonction de la proximité de la cible.

Bien que le volume peut varier de plusieurs façons, nous supposons que le volume varie à partir d'un grand A_0 (positif) à une valeur constante minimale A_{min} [25].

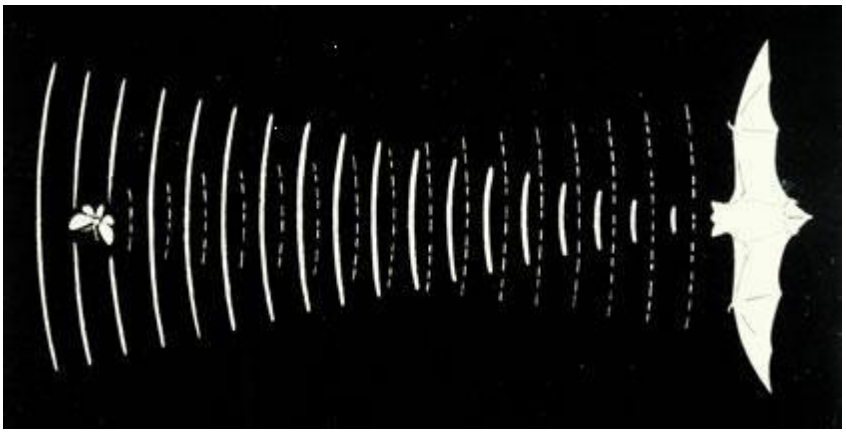

Figure 3.2: Echolocation

3.3.4 Acoustique d'écholocation

Bien que chaque pulsation ne dure que quelques millièmes de seconde (De 8 à 10 ms), elles ont des fréquences constantes qui sont habituellement dans la région 25kHz et 100 kHz pour la plus part des espèces des chauves-souris. Toutefois, certaines espèces peuvent émettre des fréquences plus élevées allant à 150 KHz [27]. Chaque cri ultrasonique peut durer moyennement 5 à 20 ms. Typiquement les microchiroptères émettent de 10 à 20 cris par seconde. En cas de chasse, le taux d'émission peut être accéléré jusqu'à environ 200 pulsations par seconde quand celles-ci volent à proximité de leurs proie. Ces cris sonores trop courts marquent la capacité étonnante du traitement puissant du signal des chauves-souris.

Comme la vitesse du son dans l'air est v = 340 m/s, la longueur d'onde λ des cris ultrasoniques avec une fréquence constante *f* est donné par

$$\gamma = \frac{V}{F}$$

Les longueurs d'onde sont du même ordre de grandeur que les tailles de leurs proies. L'impulsion émise pourrait être aussi forte que 110 dB, appartenant à la région des ultrasons. L'intensité varie également de la plus forte lors de la recherche de proie et qui diminue en la poursuivant [27]. Les microchiroptères peuvent éviter des obstacles même négligeables.
Les études ont prouvé que les microchiroptères utilisent le délai entre l'émission et la détection de l'écho, la différence de temps entre leurs deux oreilles, et les variations d'intensité des échos afin de construire un scenario tridimensionnel de l'environnement. Elles peuvent détecter la distance d'orientation de la cible, le type de la proie, et même la vitesse de mouvement de la proie telles que les petits insectes. De surcroît, les études ont suggéré que les microchiroptères sont capables de discriminer leurs cibles par la variation de Doppler effect induit par le taux de wing-flutter de l'insecte cible [28].

Bien évidemment, quelques chauves-souris ont une bonne vue, et la plupart d'entre elles ont une très grande sensibilité pour l'odorat. En réalité, elles emploient ses sens en combinaison afin de maximiser le coefficient de détection de proie et la navigation minutieuse, particulièrement en obscurité totale. Cependant nous nous intéressons uniquement au comportement d'écholocation.

Le comportement d'écholocation des microchiroptères est associé à une fonction objective à optimiser nous permettant de formuler un nouveau algorithme d'optimisation appelé Algorithme des chauves-souris.

3.3.5 Optimisation par l'algorithme des chauves-souris

Comme il a été déjà cité, l'algorithme des chauves-souris tente de modéliser le comportement d'écholocation des microchiroptères. Il repose essentiellement sur le scénario suivant: des chauves-souris volent aléatoirement en vue de trouver de la nourriture, celles-ci n'ont aucune connaissance sur l'endroit où se trouvent leurs proies, cependant elles peuvent calculer la distance qui sépare chacune d'elles de la cible au moyen de l'écholocation. Leur objectif commun est non seulement d'atteindre leurs proies mais aussi de trouver la meilleure stratégie de chasse. La stratégie la plus efficace consiste à suivre la chauve-souris la plus proche de la volée. Ce comportement est similaire à résoudre un problème d'optimisation.

L'algorithme des chauves-souris est une méthode d'optimisation ou toutes les chauves-souris collaborent en population dans le but d'atteindre un but collectif. Ce dernier est formulé par une fonction dite fonction objectif. Chacune des chauves-souris volantes est considérée comme une solution dans l'espace de recherche appelée solution candidate ou locale. Celle-ci sera évaluée à chaque itération de l'algorithme en calculant sa fitness par la fonction objectif et mise à jour en ajustant sa position, sa vélocité et sa fréquence, en fonction de la meilleure position qu'elle a atteinte et celle trouvée par toute la population. Pour que finalement l'algorithme maintienne la meilleure solution atteinte parmi toute la volée dite solution globale après un nombre d'itérations prédéfini.

Formellement, Les chauves-souris volent aléatoirement avec une vélocité V_i à une position X_i avec une fréquence fixée f_{min}, en variant la longueur d'onde λ et l'intensité A_0 de recherche de proie. Elles peuvent ajuster automatiquement la longueur d'onde (ou la fréquence) ainsi que le taux d'émission de pulsation $r \epsilon [0,1]$, selon la proximité de leurs proies.

Bien que l'intensité peut varier de plusieurs manières, l'algorithme suppose que $A \epsilon [A_0, A_{min}]$

L'algorithme des chauves-souris, est essentiellement composé des trois étapes suivantes, et qui seront répétées jusqu'à atteindre un certain nombre d'itérations :

- Evaluer la fitness de chaque chauve-souris
- Mettre à jour les meilleures solutions locales et globales.
- Mettre à jour les positions, les vélocités et les fréquences de chacune des chauves-souris.

Les deux premières étapes sont assez simples. L'évaluation de la fitness de chacune des chauves-souris s'effectue à l'aide de la fonction objectif. Tandis que la mise à jour des meilleures solutions locales et globales s'effectue en comparant la fitness courante de chacune des chauves-souris avec la fitness de la meilleure solution qu'elle a rencontré jusqu'ici, puis de choisir la meilleure d'entre elles. Quant à la recherche de la solution globale, il suffira de comparer toutes les meilleures solutions locales et garder la meilleure.

La troisième étape est la plus cruciale, en effet, le mouvement des chauves-souris est responsable de l'efficacité de l'algorithme. Les règles définissant la nouvelle solution et mettant à jour la position et la vélocité de chacune des chauve-souris dans un espace de dimensions sont les suivantes,

3.3.5 Initialisation de l'Algorithme de chauve-souris

La population initiale est générée de façon aléatoire pour n nombre de chauve-souris. Chaque individu de la population est décrit par un vecteur à valeurs réelles avec une dimension d. L'équation suivante est utilisée pour générer la population initiale.

$$\mathbf{X_{ij} = X_{min\ j} + rand(0,1)(X_{max\ j} - X_{min\ j})} \quad (17)$$

Où i = 1,2, ..., n; j = 1,2, ..., d; Xmaxj et Xminj sont les limites supérieures et inférieures pour la dimension j.

Solution, fréquence et vitesse

Dans les simulations, nous utilisons naturellement des chauves-souris virtuelles. Nous devons définir les règles de mise à jour de leurs positions xi et les vitesses vi , dans un espace de recherche bidimensionnel, à chaque itération t. Parmi toutes les solutions, il existe une meilleure solution courante x*. Les règles précédentes peuvent être traduites pour obtenir les nouvelles solutions V_i^tet vitesses V_i^t à l'étape t, par application des équations suivantes de mise à jour.

$$\mathbf{f_i = f_{min} + (f_{max} - f_{min})\beta} \quad (18)$$

$$\mathbf{V_i^t = V_i^{t-1} + \left(X_i^t - X^*\right) f_i} \quad (19)$$

$$\mathbf{X_{i=}^t X_i^{t-1}} \quad (20)$$

Où β ε [0,1] est un vecteur aléatoire issu d'une distribution uniforme. X* est la meilleure solution globale courante qui est déterminée en comparant toutes les solutions parmi tous les n chauves-souris. Alors que $\lambda_i f_i$ est l'augmentation de vitesse, nous utilisons f_i pour régler la vitesse tout en fixant l'autre facteur λ_i. La plage de valeurs de fi diffère d'un problème à l'autre en fonction du domaine, de la taille du problème, etc. Initialement, chaque chauve-souris reçoit de manière aléatoire une fréquence qui est dérivée uniformément de [f_{min}, f_{max}]. Lorsqu'une solution est sélectionnée parmi les meilleures solutions courantes, une nouvelle solution pour chaque chauve-souris est générée localement à l'aide d'une transformation intégrant un facteur aléatoire.

$$\mathbf{X_{new} = X_{old} + \varepsilon A^t} \quad (21)$$

Où ε ϵ [-1,1] est un nombre aléatoire, alors que At est le volume moyen de toutes les chauves-souris à cette étape de traitement. La mise à jour des vitesses et des positions des chauves-souris est similaire à la procédure d'optimisation standard des essaims de particules [29] [30]. Étant donné que le contrôle de la portée du mouvement des particules envahissantes, l'algorithme de chauve-souris peut être

considéré comme étant une combinaison équilibrée de l'optimisation standard des essaims de particules et de la recherche locale intensive, contrôlée par le volume et le taux de pulsation.

$$\mathbf{A_i^{t+1}} = \boldsymbol{\alpha} \mathbf{A_i^t} \quad (22)$$

$$\boldsymbol{r^{t+1}} = \boldsymbol{r_i^0}[\mathbf{1} - \mathbf{exp}(\boldsymbol{\gamma t}] \quad (23)$$

Où α et γ sont des constantes. L'α constant est similaire au facteur de refroidissement dans le recuit simulé. Pour tout 0 <α <1 et γ > 0 tel que

$$\mathbf{A_i^t} \rightarrow \mathbf{0}, \mathbf{r_i^t} \rightarrow \mathbf{r_i^0}, \mathbf{as\ t} \rightarrow \infty \quad (24)$$

Le choix des paramètres nécessite une certaine expérimentation. Chaque chauve-souris devrait avoir différentes valeurs de sonorité et taux d'émission d'impulsion. Et cela peut être réalisé par randomisation. Le volume et les taux d'émission ne sont mis à jour que si les nouvelles solutions sont améliorées, ce qui signifie que ces chauves-souris se déplacent vers la solution optimale.

3.3.7 Code de l'algorithme de chauve-souris standard.

1. Fonction fitness : f(x), x = (x1, ..., xd)
2. Initialiser la population de chauve-souris xi et la vitesse vi, i = 1,2, ..., n
3. Définir la fréquence d'impulsions fi de chaque position xi
4. Initialiser le taux de pulsation ri et le volume Ai
5. Tant que (t <nombre maximum d'itérations)

5.1 Générer de nouvelles solutions en ajustant la fréquence et en actualisant les vitesses et les positions / solutions. (équations 2, 3; 4).

5.2 . Si (rand> ri)

5.2.1 5.2.1. Sélectionnez une solution parmi les meilleures solutions

5.2.2 Générer une solution locale autour de la meilleure solution sélectionnée x* (équation 5)

5.3 Fin si

5.4 Si (rand <Ai et f(xi) <f(x*))

5.4.1 Accepter de nouvelles solutions

5.4.2 Augmenter ri et réduire Ai (équations 6, 7)

5.5 Fin si

5.6 Trouver la meilleure solution x*

6 Fin tant que

7 Afficher les résultats donnés par la meilleure solution x*

8 Fin de l'algorithme

3.4 Organigramme L'Algorithme des chauves-souris

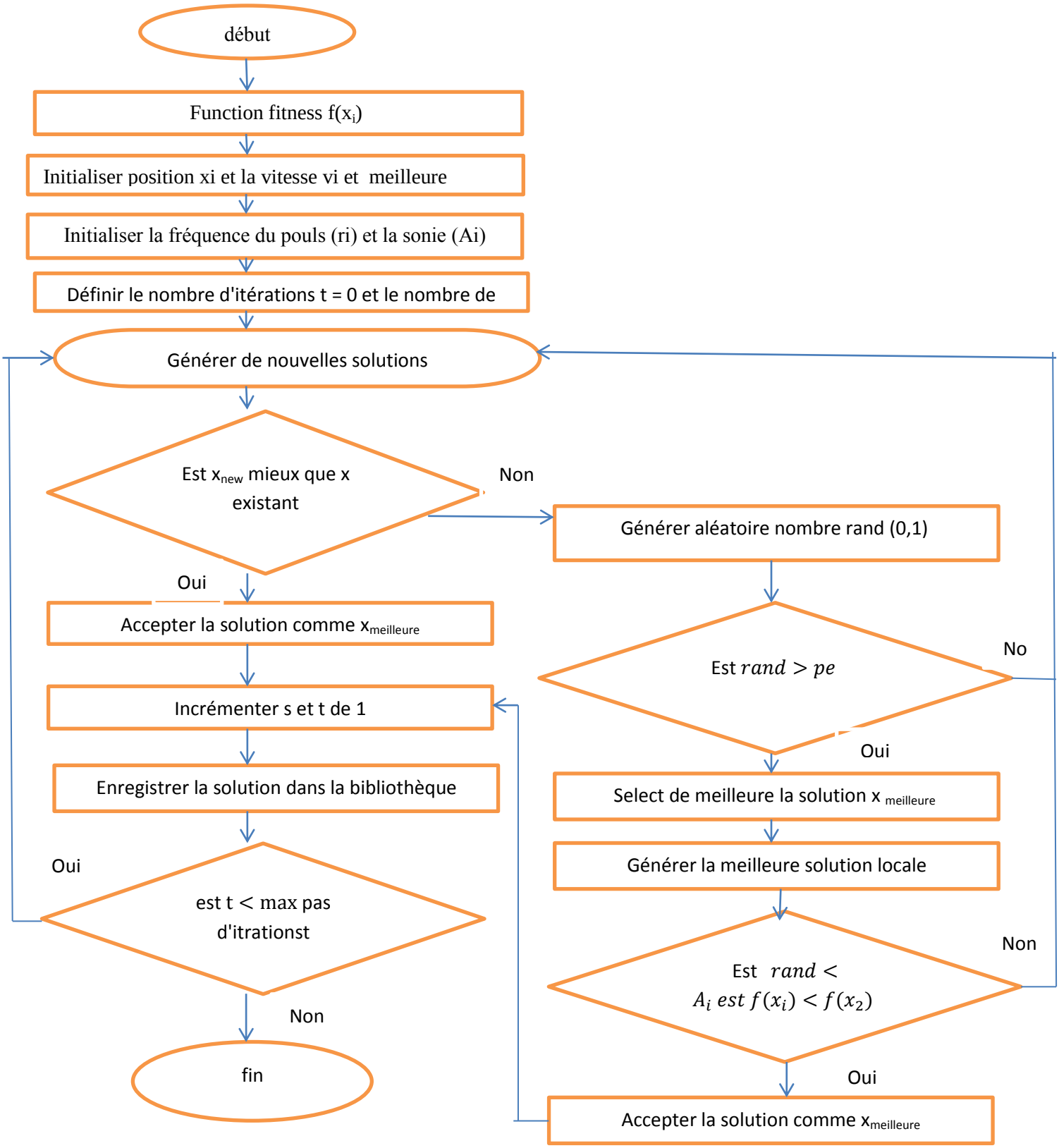

Figure 3.3: Organigramme de Algorithme des chauves-souris

3.5 Travaux connexes

3.5.1 Fast Triangle Flip Bat Algorithm (FTBA) [31]

X. J. Cai et ces collègues suggérent une stratégie de retournement de triangle pour mettre à jour la vitesse des chauves-souris. Trois stratégies différentes de retournement de triangle avec cinq conceptions différentes sont introduites. Les performances d'optimisation sont vérifiées par des benchmarks CEC2013 dans ces conceptions par rapport à la BA standard. Les résultats de la simulation montrent que la stratégie hybride de retournement de triangle est supérieure aux autres algorithmes.

3.5.2 DV Based Positioning in Ad Hoc Networks [32]

D. Niculescu, et B. Nath ont proposé l'APS un algorithme de positionnement distribué, qui fonctionne comme une extension à la fois du routage vectoriel de distance et du positionnement GPS afin de fournir une localisation approximative de tous les nœuds du réseau où une seule partie limitée des nœuds a la capacité de s'auto-identifier.

3.5.3 RoughPSO: Rough set-based particle swarm optimization [33]

L'optimisation des essaims de particules (PSO) est un algorithme d'optimisation basé sur la stochastique technique de recherche. L'ensemble grossier, en informatique, est une approximation formelle d'un ensemble conventionnel en termes de paire d'ensembles. L'ensemble approximatif donne l'approximation inférieure et supérieure de l'ensemble d'origine les propriétés de la théorie des ensembles approximative sont utilisé pour améliorer les problèmes de convergence locale dans PSO, ainsi un algorithme RoughPSO est proposé. RoughPSO utilise les ensembles d'approximations inférieure et supérieure de l'ensemble approximatif pour obtenir les valeurs d'appartenance. Ces valeurs sont ensuite utilisées pour mettre à jour la vitesse et la position de chaque particule. Une étude empirique montre que RoughPSO obtient également des taux d'exactitude de classification plus élevés sur certains ensembles de données que ceux basés sur PSO algorithmes de classification.

3.5.4 Optimal LEACH Protocol with Improved Bat Algorithm in Wireless Sensor Networks [34]

Algorithme FTBA amélioré intégrant la stratégie de courbe est proposé pour améliorer les capacités de recherche locales et globales. Ensuite, Xingjuan Cai et ces collègues combinent le BA amélioré avec LEACH et utilisent l'algorithme intelligent pour sélectionner la tête de cluster. Les résultats de l'expérience montrent que le BA amélioré a une capacité d'optimisation plus forte que les autres algorithmes d'optimisation, que la méthode qu'ils ont proposée (FTBA-TC-LEACH) est supérieure au LEACH et LEACH avec le BA standard (SBA-LEACH). Le FTBA-TC-LEACH peut évidemment réduire la consommation d'énergie du réseau et augmenter la durée de vie des réseaux de capteurs sans fil (WSN).

3.5.5 Modified Bat Algorithm for Localization of Wireless Sensor Network [35]

L'algorithme d'optimisation méta heuristique connu sous le nom d'algorithme bat est décrit afin d'évaluer la précision du problème de localisation des nœuds dans le capteur sans fil réseaux. Pendant ce temps, l'algorithme de chauve-souris existant a également été modifié en utilisant le stratégies de recherche de nourriture bactérienne de l'algorithme d'optimisation de la recherche de nourriture bactérienne. Comparé à l'algorithme de chauve-souris existant, l'algorithme de chauve-souris modifié proposé est montré par des simulations pour être constamment plus performant non seulement en augmentant les taux de réussite de localisation et vitesse de convergence rapide mais également améliorer sa robustesse.

3.6 Conclusion

Le prochain chapitre aura pour but de présenter notre L'algorithme BATDV-Hop peut être décrit comme un algorithme de localisation hybride qui combine l'algorithme DV-Hop et BA : les informations de localisation sont optimisées et transformées en coordonnées des nœuds inconnus par le BA un problème d'optimisation, puis appliquer l'algorithme des chauves-souris à ce problème, la localisation des capteurs.

Chapitre 4

Simulation BATDV-Hop

Sommaire

4.1 Introduction

On va présenter dans ce chapitre la simulation de BATDV-Hop et nous proposons une série d'expérimentations pour but de tester et évaluer les résultats de cet algorithme en terme de l'efficacité et de robustesse.

4.2 Algorithme BDV-Hop

L'algorithme BDV-Hop peut être décrit comme un algorithme de localisation hybride qui combine l'algorithme DV-Hop et BA. Premièrement, l'algorithme DV-Hop est utilisé pour acquérir les informations de localisation des nœuds en utilisant une approche sans plage. Ensuite, les informations de localisation sont optimisées et transformées en coordonnées des nœuds inconnus par le BA. Dans cet chapitre, nous supposons que les nœuds de capteurs fonctionnent comme suit.

- Tous les nœuds sont déployés au hasard dans le réseau et utilisent un protocole de routage plan.
- Dans le processus de comptage de sauts, les nœuds communiquent avec leurs nœuds voisins dans un mode de communication par inondation.
- Seuls les nœuds d'ancrage transmettent les données recueillies à partir des nœuds de capteur à la station de base; ils émettent également des ordres de la station de base aux nœuds de capteur.

4.2.1 Acquisition des informations de localisation par l'algorithme DV-Hop

Dans l'algorithme DV-Hop original, la distance entre les nœuds d'ancrage et le nœud inconnu est estimée par le produit de la distance de saut moyenne et de la valeur de comptage de sauts. Le processus d'acquisition des informations de localisation comprend trois phases[36,37].

- **Comptage de sauts**: Les nœuds d'ancrage diffusent des paquets de données contenant leurs informations de localisation et le nombre de sauts vers leurs nœuds voisins. Lorsqu'un nœud reçoit un paquet, il ajoute 1 au nombre de sauts, puis enregistre le nombre de sauts et l'emplacement du nœud d'ancrage émetteur. Par la suite, les nœuds transmettent ces paquets à leurs nœuds voisins jusqu'à ce que tous les nœuds aient reçu des paquets.
- **Estimation de la distance:** Dans la première phase, les informations de localisation et le nombre de sauts vers les nœuds d'ancrage sont enregistrés. Par conséquent, la distance de saut moyenne à un nœud d'ancrage est donnée par

$$HopSize_i = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{i \neq j} h_{ij}} \quad (25)$$

Où x_i, y_i est l'emplacement du nœud d'ancrage i et h_{ij} est le nombre de sauts entre le nœud d'ancrage i et le nœud d'ancrage j. La distance estimée est calculée par

$$d_{jk} = HopSize_i \text{x} h_{ij} \quad (26)$$

Où d_{jk} est la distance estimée entre le nœud d'ancrage i et le nœud k.

(3) Calcul des coordonnées. Dans (25) et (26), l'emplacement du nœud inconnu est exprimé par

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1 \\ (x_2 - x)^2 + (y_2 - y)^2 = d_2 \\ \quad - \\ \quad - \\ (x_n - x)^2 + (y_n - y)^2 = d_n \end{cases} \quad (27)$$

Où(x, y)est la coordonnée d'un nœud inconnu. Ensuite, (27) est transformé en forme matricielle

$$AX = b \quad (28)$$

$$\text{A}=\begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ 2(x_2 - x_n) & 2(y_2 - y_n) \\ & . \\ & . \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix}, X = \begin{bmatrix} x \\ y \end{bmatrix} etb = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 - d_1^2 + d_n^2 \\ x_2^2 - x_n^2 + y_2^2 - y_n^2 - d_2^2 + d_n^2 \\ . \\ . \\ x_{n-1}^2 - x_n^2 + y_{n-2}^2 - y_n^2 - d_{n-1}^2 + d_n^2 \end{bmatrix}$$

Alors, X est représenté par

$$\text{X} = (A^t A)^{-1} A^t b \quad (29)$$

4.2.2 Optimisation et transformation des informations de localisation par la BA

La solution optimale pour (29) est obtenue en résolvant un problème d'optimisation multi-objectifs. L'erreur de distance est définie comme

$$|d_i^* - d_i| \leq \varepsilon_i \tag{30}$$

Où d_i désigne la distance estimée entre le nœud d'ancrage i et le nœud inconnu et d_i^* désigne la distance euclidienne réelle entre le nœud d'ancrage i et le nœud inconnu.

L'erreur de distance globale ε entre les nœuds d'ancrage $(xi\,, yi)$ et le nœud inconnu $(xi\,, yi)$ peut être exprimée comme

$$\begin{cases} d_1^2 - \varepsilon_1^2 \leq (x - x_1)^2 + (y - y_1)^2 \leq\ d_1^2 + \varepsilon_1^2 \\ d_2^2 - \varepsilon_2^2 \leq (x - x_2)^2 + (y - y_2)^2 \leq d_2^2 + \varepsilon_2^2 \\ \qquad\qquad - \\ \qquad\qquad - \\ d_n^2 - \varepsilon_n^2 \leq (x - x_n)^2 + (y - y_n)^2 \leq d_n^2 + \varepsilon_n^2 \end{cases} \tag{31}$$

De toute évidence, la précision des emplacements estimés en (30)et (31)est corrélée négativement avec l'erreur de distance. Pour évaluer la précision de la localisation, une fonction de fitness est modélisée comme

$$F(x,y) = \sum_{i=1}^{n} (\sqrt{(x - x_i)^2 + (y - y_i)^2} - d_i)^2 \tag{32}$$

Dans le BA, l'emplacement d'une chauve-souris représente une des solutions potentielles pour (32). Par conséquent, la résolution du problème de l'optimisation multi-objectifs est considérée comme la recherche des emplacements individuels optimaux du groupe de chauves-souris. Le processus d'optimisation et de transformation des informations de localisation d'un nœud inconnu par le BA est divisé en quatre étapes. [38, 39].

- Initialisez le groupe de chauves-souris. Le groupe de chauves-souris est généré de manière aléatoire et chaque chauve-souris est initialisée avec un emplacement et une vitesse. Ensuite, le groupe de chauves-souris est évalué par la fonction de fitness dans (32). La chauve-souris avec la plus petite valeur de fitness est choisie comme étant l'individu optimal initial, et son emplacement est la solution optimale initiale $x*$
- Mettez à jour le statut du groupe de chauves-souris. L'état de chaque batte est mis à jour en ajustant la fréquence d'impulsion comme

$$\begin{cases} f_i = f_{min} + (f_{max} - f_{min})\beta & a \\ v_i^t = v_i^{t-1} + (x_i^t - x^*) f_i & b \\ x_i^t = x_i^{t-1} + v_i^t & c \end{cases} \tag{33}$$

Où f_i est la fréquence d'impulsion de la batte i, f_{max}est la fréquence d'impulsion maximale, f_{min}est la fréquence d'impulsion minimum, $\beta(\beta \in [0,1])$est un facteur aléatoire avec unedistribution uniforme, v_i^test la vitesse de la bat$i$au temps$t$, et x_i^test l'emplacement de la batte$i$.

- Mettez à jour les solutions. Une nouvelle solution avec une perturbation aléatoire est générée avec une probabilité donnée par

$$x_i^{'} = x_i + \alpha A^t \quad (34)$$

Où $\alpha(\alpha \in [-1,1]$est une variable aléatoire,et A^test le paramètre de volume de bat i au temps t.

Ensuite, comparez les valeurs de fitness de la nouvelle solution$x_i^{'}$avec la solution optimale précédente. Si$x_i^{'}$est plus petit, $x_i^{'}$remplace la solution optimale précédente. Enfin, mettez à jour les solutions du groupe bat et continuez à rechercher de meilleures solutions optimales.

- Ajustez les paramètres de manière adaptative. Selon les principes biologiques du comportement de recherche de chauve-souris, sa fréquence d'impulsion est faible mais son volume est élevé pendant le processus de recherche initial, ce qui conduit à élargir la portée de la recherche. Plus tard, la chauve-souris augmente la fréquence des impulsions et diminue progressivement le volume pour améliorer la précision de la recherche. Dans le BA, la fréquence et le volume d'impulsion sont mis à jour par

$$A_i^{t+1} = \delta A_i^t \quad (35)$$

Où $\delta(0 < \delta < 1$ et $\gamma(\gamma > 0)$sont des constantes, r_i^{t+1}est le taux d'émission d'impulsions autemps $t+1$, et r_i^0est le taux d'émission d'impulsions maximum.

Répétez les trois dernières étapes jusqu'à ce que la condition de terminaison itérative soit atteinte. La dernière solution optimale est l'emplacement du nœud inconnu.

On va présenter la partie d'implémentation de l'algorithme chauve souris et nous proposons une série d'expérimentations pour but de tester et évaluer les résultats de cet algorithme en terme de l'efficacité et de robustesse.

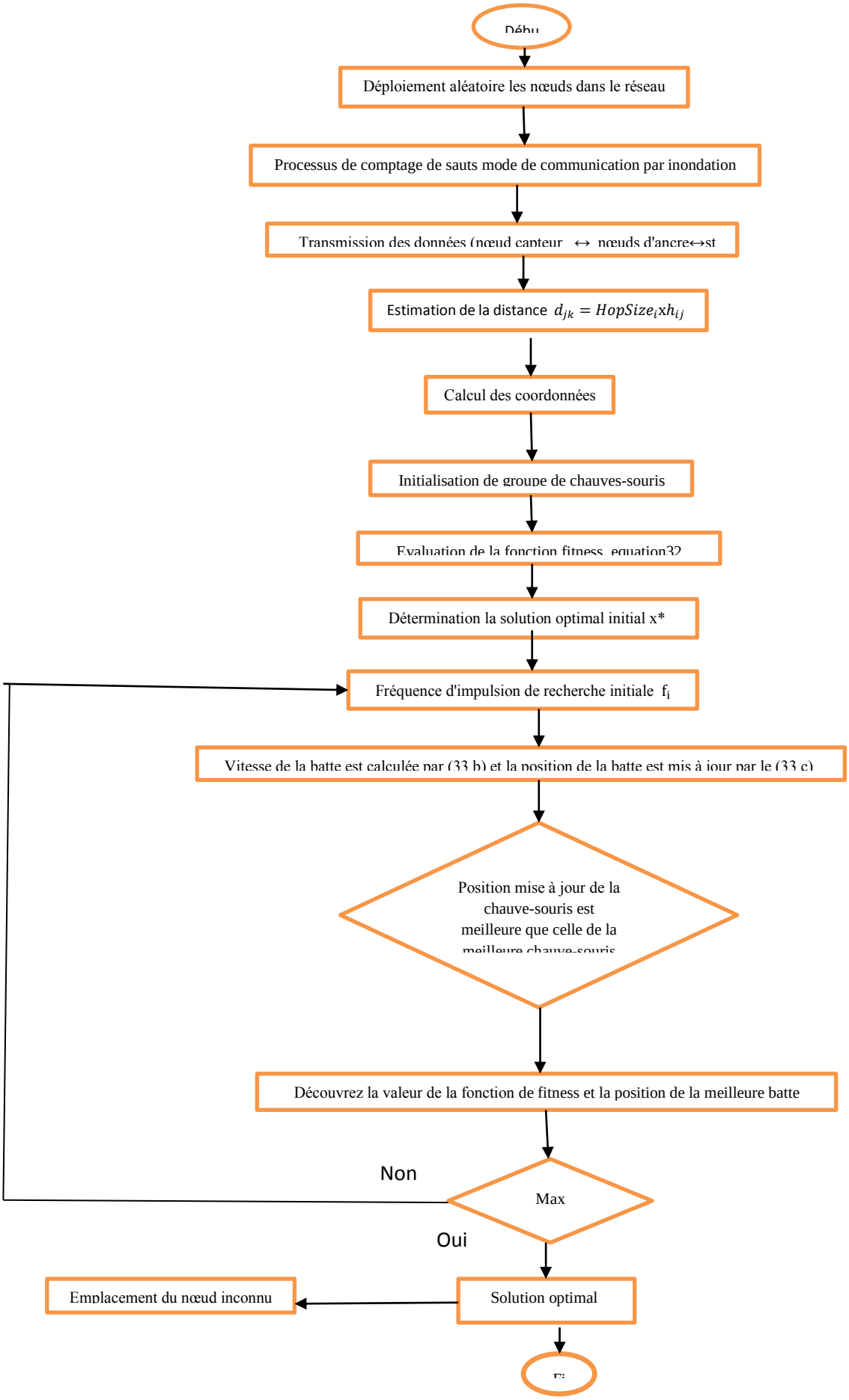

Figure 4.1 Organigaramme de algorithme de BATDV-Hop

4.3 Choix du langage de programmation

Nous avons développé notre application à l'aide du langage MATLAB version R2012a sur Windows 7 Ultimate 32 bit, de RAM de 4.00 Go, et de processus Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz, 2.60GHz.

4.3.1 Bref présentation de MATLAB

MATLAB (Matrix Laboratory) est un environnement informatique multi-paradigme et un langage de programmation de quatrième génération. Un langage développé par Math Works, MATLAB permet des manipulations matricielles, le traçage des fonctions et des données, la mise en œuvre d'algorithmes, la création d'interfaces utilisateur et l'interface avec des programme **.**

4.3.2 Pourquoi le choix du langage MATLAB

Nous choisissons le langage MATLAB pour les raisons suivants:

MATLAB est un language de haut niveau pour le calcul scientifique et technique.

- Environnement bureau pensé pour l'exploration itérative, la conception et la résolution de problèmes.
- Graphiques destinés à la visualisation de données et outils conçus pour créer des tracés personnalisés.
- Applications dédiées à l'ajustement de courbes, la classification de données, l'analyse de signaux et bien d'autres tâches spécialisées.
- Boîtes à outils additionnelles conçues pour répondre à de nombreux besoins spécifiques aux ingénieurs et aux scientifiques.
- Outils permettant la création d'applications avec interface utilisateur personnalisée.
- Interfaces vers C, C++, FORTRAN, Java™, COM, et Microsoft® Excel®.
- Options de déploiement libre de droits permettant de partager des programmes MATLAB avec les utilisateurs finaux.
- Le principe raison de choix de cette langage est que notre application est basée essentiellement sur les calculs mathématiques, particulièrement les calculs matricielles , et MATLAB est riche de fonctions mathématiques prédéfinies, ce qui rend l'implémentation de notre projet plus facile qu'on utilise d'autres langages de programmation

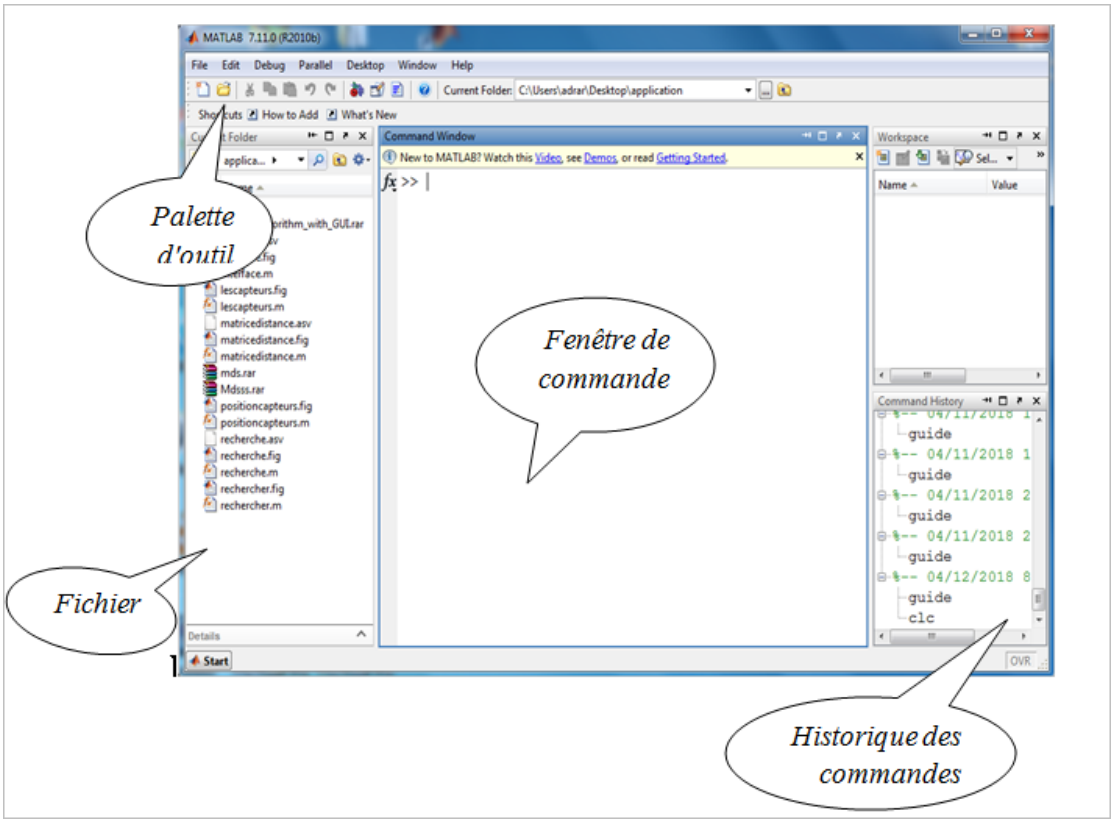

Figure 4.2: Interface du MATLAB

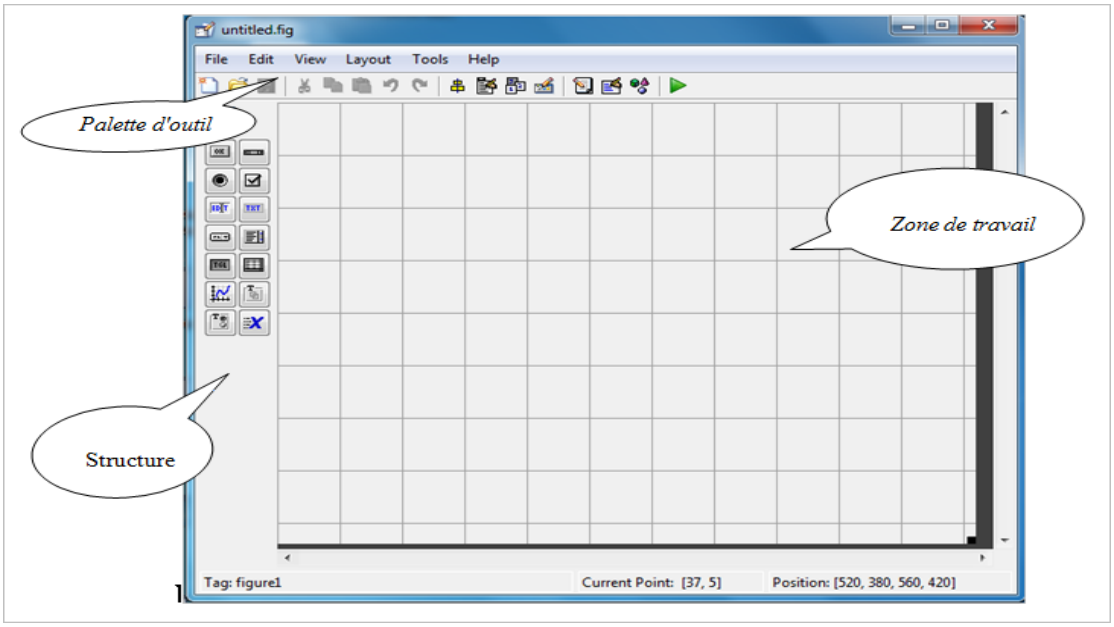

Figure 4.3: Fenêtre GUI en MATLAB

4.4 Description des étapes de simulation

L'interface graphique de cette application est comme suit :

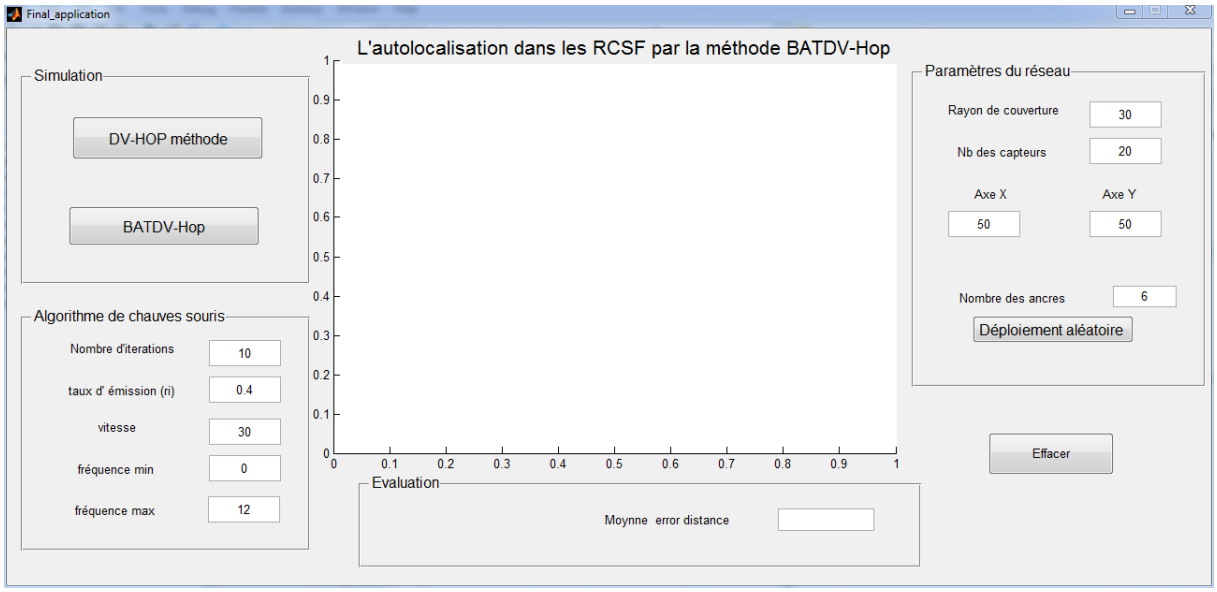

Figure 4.4: Interface de simulation.

4.4.1 Choix des paramètres du réseau

Avant de simuler l'algorithme de DV-Hop, if faut d'abord saisir tous les paramètres de réseau définis, à savoir:

- Le rayon de couverture.
- Nombre des capteurs.
- Les coordonnées (x,y) de la zone de simulation.
- Nombre des ancres.

Tous ces paramètres doivent être saisis pour que l'algorithme DV-Hop puisse fonctionner.

4.4.2 Déploiement des capteurs

Après de saisir les paramètres de simulation, on fait le déploiement aléatoire de capteurs dans la zone de simulation, c'est la première tâche effectuée avant la localisation. Chaque capteur prend un emplacement (x,y) purement aléatoire de tel sorte que chaque capteur occupe une position propre différente des autres capteurs.

4.5 Localiser par DV-hop

Une fois les capteurs sont déployés aléatoirement, la localisation par la méthode DV-hop doit être effectuée

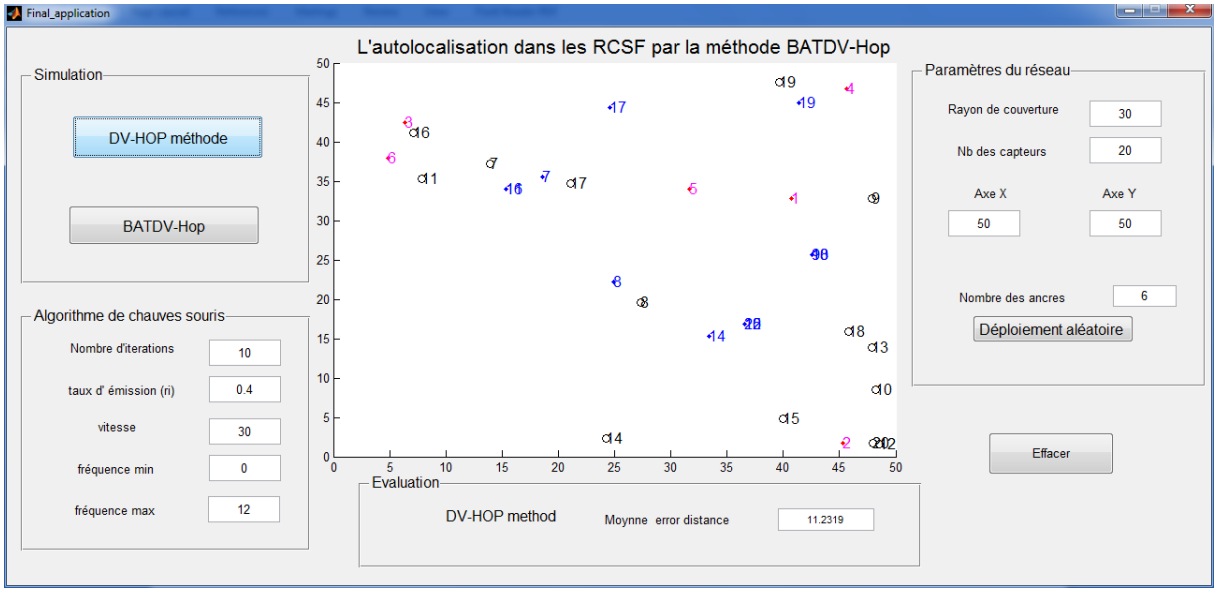

Figure 4.5: Localisation par la méthode DV-hop

Pour exécuter l'algorithme BATDV-Hop, en plus des paramètres cités au-dessus, nous choisissons les paramètres de l'algorithme de BAT à savoir:

- Nombre d'itération
- Taux d'émission
- Vitesse
- Fréquence min
- Fréquence max

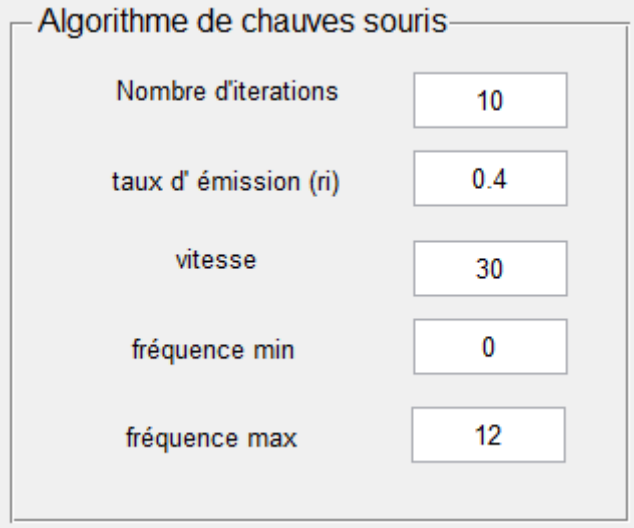

Figure 4.6: Paramètres de l'algorithme de chauves-souris

4.5.1 Paramètres de simulation

Tableau 4.1: Paramètres de simulation de algorithme DV-Hop

Paramètres	**Valeur**
Nombre des capteur	20
Rayon de couverteur	30
Zone de simulation	(50,50)
Nombre de nœuds d' oncre	6

Tableau 4.2: Paramètres de simulation d'algorithme de chauves-souris.

Paramètres	**Valeur**
Nombre d'itération	5
Taux d'émission ri	0.4
Vitesse	30
Fréquence min	0
Fréquence min	12

4.5.2 Métrique d'évaluation

Afin d'évaluer les deux algorithme: DV-Hop et BATDVHop, nons avons considére AED (Average Ereur Distance)comme méthode d'évaluation pour faine la comparaison enter ce deux algorithmes

4.5.3 Résultats

Quatre scénarios ont été simulés pour tester l'efficacité de l'algorithme DV-Hop, dont nous avons varié le nombre des ancres et/ou le nombre des nœuds dans chaque scénario.

Tableau 4.3 : Résultats de simulation

DV-Hop	Nombre de capteur 20 Nombre d'ancre: 6	Nombre de capteur 30 Nombre d'ancre: 6	Nombre de capteur 20 Nombre d'ancre: 3	Nombre de capteur 30 Nombre d'ancre: 3
Simulation 1	11,2319	8,109	20,1601	70,0352
Simulation 2	10,9753	13,923	14,8135	65,1245
Simulation 3	10,7308	7,8187	9,2055	61,1512
Simulation 4	9,265	9,1471	21,0451	27,2116
Simulation 5	10,0079	11,7725	12,7399	16,9989
Moyenne	10,44218	10,15406	15,59282	47,80428

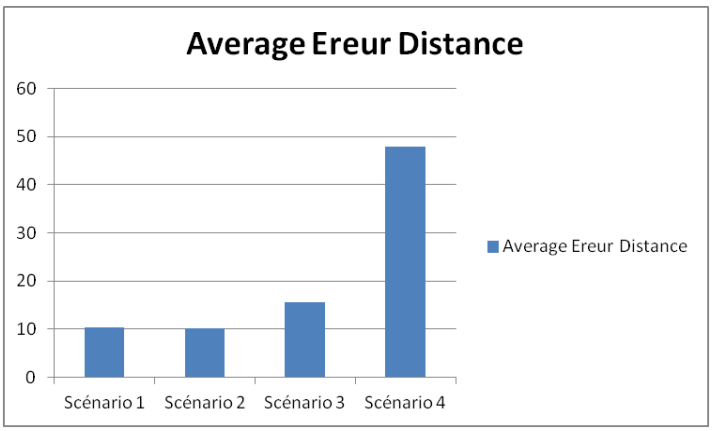

Figure 4.7 : Distance moyenne erreur de quatre scénarios

D'après la figure précédente, nous pouvons constater que le nombre des ancres a une influence directe sur la précision des résultats. Plus le nombre des ancres est élevé, plus le taux d'erreur est faible et vice versa.

4.6 Conclusion:

Dans ce chapitre, nous avons implémenté l'algorithme de DV-Hop afin de trouver / estimer les coordonnées inconnus des nœuds et nous avons essayé d'implanter l'algorithme BATDV-Hop proposé. D'après les résultats obtenus, nous avons déduit que le nombre des ancres démunie le taux d'erreur.

Conclusion générale

Les réseaux de capteurs sans fil ont un large potentiel et constituent un sujet de recherche innovant ainsi qu'un outil souhaité par plusieurs domaines. C'est sans aucun doute, une technologie qui va nous accompagner pour les prochaines années et ainsi faire partie de notre vie quotidienne. Cependant, il y a encore beaucoup de problèmes qui doivent être abordés pour un fonctionnement efficace de ces réseaux dans des applications réelles. Parmi les problèmes fondamentaux et importants dans ces réseaux de capteurs nous citons le problème de la localisation qui est une n´nécessité absolue à laquelle des solutions adéquates doivent être proposées.

Le travail consigné dans ce mémoire a été le fruit d'une étude menée dans le contexte des réseaux de capteurs sans fil, ce qui nous a permis de découvrir les propriétés de ces derniers, leurs contraintes et des domaines variés qui les utilisent. Nous nous sommes intéresses principalement à la problématique de la localisation en ´étudiant la position des capteurs et en prenant d'autre métrique comme la précision de localisation.
Du moment que l'information de positionnement fait partie de la plupart des services des réseaux de capteurs, tels que le routage géographique et la surveillance. . .. Il est nécessaire de proposé un algorithme de localisation. Notre protocole est base sur la méthode BATDV-Hop pour le calcul des distances et pour le calcul de position.

Dans ce mémoire, nous avons mis en place une méthode de localisation afin d'estimer les positions (coordonnées) des nœuds en utilisant une méthode requiert l'encodage au préalable de la position d'un certain nombre d'ancres. Nous avons proposé un algorithme nommé BATDV-Hop base sur l'hybridation entre algorithme DV-Hop et algorithme de chauves-souris.

Comme perspective, nous allons continuer à améliorer ce travail, et surtout en prenant d'autres métriques comme la consommation d'énergie, la scalabilité, et la précision de localisation. Nous envisageons tous cela dans nos futurs travaux de recherche qu'ils soient d'ordre individuels, professionnels ou académiques.

Bibliographie

[1] 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.

[2] V.Ailawadhi K.Sohrabi, J.Gao and G.Pottie. Protocols for self-organization of a wirelesssensor network. volume 7, pages 16–27. IEEE Communication Magazine, 2000

[3] V. Raghunathan ; C. Schurgers ; S. Park ; and M. B. Srivastava. Energy-aware wireless microsensor networks. IEEE Signal Processing Magazine

[4] MAAROUF, Samia, and Souhila OUADAH. "Implémentation et évaluation des schémas de routage sur une plateforme réelle de réseaux de capteurs sans fil." PhD diss., 2014

[5] Abdallah, M. A. K. H. O. U. L., J. Bahi, and A. Mostefaoui. "Réseaux de capteurs: localisation, couverture et fusion de données." PhD diss., Thèse de Doctorat, Université de Franche-Comté, 2008.

[6] Hill, Jason Lester. "System architecture for wireless sensor networks." PhD diss., University of California, Berkeley, 2003.

[7] S. Sudevalayam, P. Kulkarni, 'Energy harvesting sensor nodes: Survey and implications. Communications Surveys & Tutorials', pages 1–19, 2010.

[8] S. Jain. Energy aware communication in ad-hoc networks. Technical Report UW-CSE, 2003.

[9] Quande Dong, XuXu , A Novel weighted centroid localization algorithme based on RSSI for an outdoor environment, journal of communication vol.9,no. 3, march 2014.

[10] http://fr.wikipedia.org. (Accès juin 2015

[11] Messaoud Belloula, La géolocalisation dans les réseaux de capteurs sans fil ; Etude de cas utilisation en agriculture. Thèse de Magistère ; Université Hadj Lakhder-Batna, 2012

[12] A.Boukerche. Algorithms And Protocols For Wireless Sensor Networks. Wiley-IEEE Press, 1 edition, 2008.

[13] A.Willig. Protocols And Architectures For Wireless Sensor Networks. Wiley edition, april 2005.

[14] E.Kim I.Wha Hong J.Lee, W.Chung. Robust dv-hop algorithm for localization in wireless sensor network. pages 2506 – 2509. Control Automation and Systems (ICCAS)

[15] F.Caixin ; Q.Zhihong ; J.Guang ; and W.Xue Z.Yanhang. An improved dv-hop localization algorithm in wireless sensor network. Information Technology and Applications (ITA), 2013 International Conference Year=2013 , Pages= 13– 16,

[16] L.Shaoqiang ; Z.Yuan ; Z.Zhi ; L.Yongzhou ; and F.Xiaoping. Improved dv-hop algorithm for high accuracy and low energy consumption. volume 3, pages 349–353, 2012.

[17] F.Yu B.Zhang. Low-complex energy-efficient localisation algorithm for wireless sensor networks using directional antenna. volume 4, pages 1617–1623, 2010.

[18] L.Chen F.Dai, Y.Liu. A hybrid localization algorithm for improving accuracy based on rssi/aoa in wireless network. pages 631 – 634. Computer Science et Service System (CSSS), 2012 International Conference, 2012

[19] S. Gezici ; Z. Sahinoglu. Enhanced position estimation via node cooperation. Cape Town, South Africa, May 2010. in Proc. IEEE International Conference on Communications (ICC).

[20] H. Wymeersch ; J. Lien ; and M. Z. Win. Cooperative localization in wireless networks. volume 97 of 2, pages 427–450. Proceedings of the IEEE, February 2009.

[21] X. J. Cai, H. Wang, Z. H. Cui, J. H. Cai, Y. Xue, and L. Wang, "Bat algorithm with triangle-flipping strategy for numerical optimization," International Journal of Machine Learning and Cybernetics, vol. 9, no. 2, pp. 199-215, 2018.

[22] D. Niculescu, and B. Nath, "DV based positioning in ad hoc networks," Telecommunication Systems, vol. 22 no. 1-4, pp. 267-280, 2003

[23] J. C. Fan, Y. Li, L. Y. Tang and G. K. Wu, "RoughPSO: rough set-based particle swarm optimisation," International Journal of Bio-Inspired Computation, vol. 12, no. 4, pp. 245-253, 2018.

[24] X. J. Cai, Y. Sun, and Z. H. Cui, "Optimal LEACH Protocol with Improved Bat Algorithm in Wireless Sensor Networks," KSII Transcactions on Internet and Information Systems, vol. 13, no. 5, pp. 2469-2490, 2019.

[25] Sonia Goyal, Manjeet Singh Patterh, " Modified Bat Algorithm for Localization of Wireless Sensor Network," Published online: 25 July 2015