

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 : Système intelligents

Thème

*Application de l'essaim de poulet pour la
localisation dans les RCSFs*

Préparé par

CHETOUAH Karima & BADDOU Saliha

Devant le jury :

Encadreur : Mr. RABHI Seddiki

Président : Dr. OMARI Mohammed

Examineur1 : Mr. MAMOUNI Elmamoun

Examineur2 : Mr. DEMRI Mohammed

Année Universitaire 2018/2019

Résumé :

Le réseau de capteurs sans fil est composé d'un grand nombre de nœuds qui sont déployés de manière dense dans une région d'intérêts pour mesurer certains phénomènes. L'objectif principal est de déterminer l'emplacement du nœud du capteur. Récemment, les techniques de localisation bio-inspirées sont devenues populaires grâce à sa solution précise et rapide. Dans ce mémoire, un algorithme méta-heuristique récemment développé basé sur le comportement social des poulets appelé optimisation par essaims de poulets (CSO) est proposé pour résoudre le problème de localisation des nœuds RCSFs. De plus, on a proposé une version amélioré de CSO pour localiser les nœuds qui ont moins de trois ancrs dans le voisinage. La simulation et l'étude expérimentale effectuée montrent l'efficacité de nos algorithmes proposés.

Mot clés : Réseaux de capteurs sans fils, RCSFs, localisation, bio-inspirées, méta-heuristique, optimisation des essaims de poulets, CSO.

Abstract:

Wireless sensor network is composed of a large number of nodes that are densely deployed in a region of interest to measure certain phenomena. The main objective is to determine the location of the sensor node. Recently, bio-inspired localization techniques have become popular thanks to its accurate and fast solution. In this thesis, a recently developed meta-heuristic algorithm based on the social behavior of chickens called chick swarm optimization (CSO) is proposed to solve the problem of localization of RCSF nodes. In addition, an improved version of CSO has been proposed to locate nodes that have fewer than three anchors in the vicinity. The simulation and the experimental study carried out show the effectiveness of our proposed algorithms.

Key words: Wireless sensor networks, WSN, localization, bio-inspired, meta-heuristic, optimization of chicken swarms, CSO.

ملخص:

شبكة الاستشعار اللاسلكية (WSN) هي عدد كبير من العقد التي يتم نشرها بكثافة في منطقة ذات أهمية لقياس بعض الظواهر. الهدف الأساسي هو تحديد موقع عقدة المستشعر. في الأونة الأخيرة ، أصبحت تقنيات Bioinspired في التوطن شائعة بسبب حلها الدقيق والأسرع .

في هذا العمل ، تم اقتراح خوارزمية استدلالية مطورة حديثاً تستند إلى السلوك الاجتماعي للدجاج المسمى تحسين سرب الدجاج (CSO) لحل مشكلة توطن عقدة RCSF بالإضافة إلى ذلك ، تم اقتراح نسخة محسنة من CSO لتحديد العقد التي تحتوي على أقل من ثلاثة نقاط ربط في المنطقة المجاورة. أظهرت المحاكاة والدراسة التجريبية فعالية الخوارزميات المقترحة

الكلمات المفتاحية : شبكات الاستشعار اللاسلكية , WSN ، تحديد المواقع , استدلالية, Bioinspired , تحسين سرب الدجاج, CSO .



Dédicace

C'est avec grand plaisir que je dédie ce modeste travail :

A mes très chers parents, qui m'ont soutenu, encouragé pour que je puisse mener à bien mes études, et qui attendu ce jour avec impatience.

A tous mes frères, mes sœurs et toute ma famille, en témoignage de mon profond respect

A mes amis et collègues, et tous ceux qui m'ont aidé durant le parcours de mes études.

A mon binôme : Karima.

A mes enseignants et mes amies de l'étude.

A tous ceux que j'aime dans le monde.



Salika



Dédicace

Cette étude est dédiée à ma

Mère bien-aimée, qui a été ma source d'inspiration

Et ma force lorsque j'ai pensé abandonner,

Qui fournissent continuellement son soutien

Moral, spirituel, émotionnel et financier.

À mes frères Khaled, Rabie, Toufik, Bilal et Walid, et mes amis sur tout

Chahira et Nouera et camarades de classe

qui ont partagé leurs conseils et leurs encouragements

pour terminer cette

Étude. Je dédie mon travail à

L'âme de mon père.

KARIMA.

Remerciements

Avant tout

Le grand et vrai merci à الله qui nous avons donné la force et la vie pour mener à terme ce projet.

nous tiens à remercier en premier lieu Mr. RABHI Seddik , notre encadreur de notre thèse pour ses conseils et pour son aide qui nous a conduits à concrétiser ce travail.

Nous remercions également tous les membres du jury d'avoir accepté à participer à l'évaluation de notre travail.

Nos vifs remerciements envers toutes les personnes qui ont contribué au bon déroulement de ce travail.

Enfin, nous adressons notre plus sincères remerciements à tous notre proches et amis, qui nous ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire.

Table des matières

Résumé	I
Abstract	II
Dédicaces	III
Remerciements	V
Table des matières	VI
Liste des figures	X
Liste des abréviations	XIII
Introduction générale	XIV

Chapitre I : Généralités Sur Les Réseaux De Capteurs Sans Fil

1. Introduction.....	2
2. Définitions	2
2.1 Un capteur	2
2.2 Définition d'un RCSF	2
3. Architecture et composant d'un capteur sans fil.....	3
3.1 L'unité de détection	4
3.2 L'unité de traitement	4
3.3 L'unité de communication	4
3.4 L'unité énergie	4
3.5 L'unité de stockage	5
4. Type de capteurs	5
5. L'architecture de communication d'un RCSF	6
6. Pile de protoculaire	6
6.1 Rôle des couches	7
6.1.1 La couche physique	7
6.1.2 La couche liaison	7
6.1.3 La couche réseau	7
6.1.4 La couche transport	8
6.1.5 La couche application	8
6.2 Plans des gestions.....	8
6.2.1 Le plan de gestion d'énergie	8
6.2.2 Le plan de gestion de mobilité	8
6.2.3 Le plan de gestion de tâche	8
7. Caractéristiques d'un RCSF	8
8. Topologie des RCSF	9
8.1 Topologie Hiérarchique	9

8.2 Topologie plate	10
9. Couvertures dans les RCSF	10
9.1 Couverture	10
9.2 Couverture d'un point	10
9.3 Couverture d'une zone	10
10. Connectivité dans les RCSF	11
11. Domaines D'application des RCSF	11
11.1 Domaine militaire	11
11.2 Domaine médicale	11
11.3 Domaine de transport	11
11.4 Domaine civil	12
12. Contraintes de RCSF	12
12.1 Durée de vie du réseau	12
12.2 Ressources limitées	12
12.3 Bande passante limitée	13
12.4 Facteur d'échelle	13
12.5 Topologie dynamique	13
12.6 Agrégation de donnée	13
13. Conclusion	13

Chapitre II : Localisation dans les réseaux de capteurs

1. Introduction.....	15
2. Définitions	15
3. Pourquoi la localisation ?	15
4. Le processus de la localisation dans les RCSF	16
5. Réseaux de capteurs et localisation	17
5.1 La localisation dans les RCSF	17
5.1.1 Estimation des distances	17
5.1.1 Dérivation des positions	17
5.2 Caractérisation des méthodes	17
5.2.1 Utilisation d'estimations de distances	17
5.2.1.1 Les méthodes range-free	17
5.2.1.2 Les méthodes range-based	18
5.2.2 Nécessité de connaître la position d'ancres	18
5.2.2.1 Les méthodes anchor-based	18
5.2.2.2 Les méthodes anchor-free	18

5.2.3	Forme d'implémentation	18
5.2.3.1	Les méthodes centralisées	18
5.2.3.2	Les méthodes distribuées	18
5.3	Critères de localisation	19
6.	Estimation des distances	19
6.1	Indicateur de puissance du signal reçu (RSSI).....	20
6.2	Angles d'arrivée (Angle of Arrival (AoA)):	20
6.3	La différence du temps d'arrivée (TDoA).....	20
7.	Dérivation des positions	20
7.1	La trilatération	20
7.2	La triangulation	21
8.	Technologie de localisation	21
8.1	Infrarouges	22
8.2	GPS : Global positioning system	22
8.3	RF-WIFI, RF-Bluetooth :	22
8.4	Image	22
8.5	Ultra (sons)	22
9.	Algorithmes de localisation	23
9.1	La méthode DV- Euclidean	23
9.2	La méthode APIT	24
10.	Conclusion	25
Chapitre III : Optimisation par Essaim poulet		
1.	Introduction.....	27
2.	Problème d'optimisation	27
3.	Définition de Méta heuristique d'optimisation	27
4.	Optimisation des essaims de poulet	27
4.1	Biologie générale	28
4.1.1	La hiérarchie chez les poules	28
4.1.2	Les différents comportements hiérarchiques	29
4.1.3	La hiérarchie remise en question	29
4.2	Principe de CSO	30
4.3	Méthodologie de division de groupe	31
4.4	Mouvement des poulets	32
4.4.1	Mouvement des cops	32
4.4.2	Mouvement des poulets.....	33

4.4.3 Mouvement des poussins	33
4.5 es étapes CSO.....	34
4.6L'algorithme de CSO	35
5. Conclusion.....	36

Chapitre IV: Implémentation et Discussion

1. Introduction.....	38
2. Le langage de programmation utilisé	38
3. Description générale de l'algorithme proposé dans RCSF	39
4. Algorithme propose	43
5. Les étapes de la simulation	44
6. Description des étapes d'exécution d'application	45
6.1 Choix des paramètres de réseau	45
6.2 Déploiement des capteurs	45
6.3 Choix du paramètre du CSO	45
6.4 Localiser par CSO	46
6.5 L'Affichage des informations supplémentaire	47
7. Évaluation l'algorithme de CSO	47
7.1 La précision de l'algorithme	48
7.2 Technique d'évaluation	48
8. Comparaison et analyse	52
9. Conclusion	54
Introduction générale	XVI

Liste de figure :

Figure I.1 : Exemple d'un Capteur	3
Figure I.4 : réseaux de capteur sans fil	4
Figure I.2 : Architecteur d'un Capteur	4
Figure I.3 : Évolution des capteurs	6
Figure I.5 : Architecteur d'un réseaux Capteur sans fil	7
Figure I.6 : La pile protocolaire de communication	8
Figure I.7 : Topologie hiérarchique	10
Figure I.8 : Topologie plate	11
Figure I.9 : Topologie base localisation	12
Figure I.10 : Quelques domaines d'applications des RCSF	14
Figure II.1: processus de localisation dans les réseaux capteurs.....	19
Figure II.2: Graphe représentant la Trilatération	23
Figure II.3 : Graphe représentant la triangulation	24
Figure II.4 : Algorithme DV-Euclidean	26
Figure II.5 : La méthode APIT	27
Figure III.1 : comportement poulet dans la nature	31
Figure III.2 : Comportement des essaims de poulet	32
Figure III.3 : système hiérarchique de l'essaim de poules	32
Figure III.4 : L'organigramme de l'algorithme CSO	36
Figure IV.1 : L'interface du MATLAB.	39
Figure IV.2 : L'interface de simulation.....	39

Figure IV.3 : Erreur de localisation par l’algorithme CSO rapport le nombre de capteurs ...	39
Figure IV.4 : Erreur de localisation par l’algorithme CSO par rapport le rayon de connectivite	39
Figure IV.5 : Erreur de localisation par l’algorithme CSO rapport le nombre des ancrs	39
Figure IV.6 : Erreur de localisation par l’algorithme CSO par rapport la taille de population	39
Figure IV.7 : Erreur de localisation par l’algorithme CSO par rapport le nombre d’itération	39
Figure IV.8 : L’interface du MATLAB.	39
Figure IV.9 : L’interface du MATLAB.	39
Figure IV.10 : L’interface du MATLAB.	39

La liste des abréviations

RCSF	Réseaux de Capteurs Sans Fil
GPS	Global Positioning System
DSP	Digital Signal Processors
FPGA	Field Programmable GateArray
ASIC	Application Specific Integrated Circuit
RF	Radio Fréquence
EUI	End-system Unique Identifier
LDA	Location Dependent Address
RSSI	Received Signal Strength Indicator
TDOA	Time Difference of Arrival
AOA	Angle of Arrival.
WIFI	Wireless Fidelity
APIT	Approximate Point In Triangulation
CSO	Chicken Swarm Optimization

Introduction générale :

Ces dernières années, l'intérêt porté aux applications des réseaux de capteurs sans fil (RCSFs) est croissant. Dans la plupart de ces applications, les informations de localisation de capteurs, qui prennent en charge de nombreux autres services réseau, sont d'une importance capitale.

L'objectif principal de la localisation de capteur est d'imputer un emplacement spécifique aux périphériques inconnus dans la zone d'intérêt. La localisation est effectuée sur la base des appareils de mesure de localisation tels que GPS (Global Positioning System). Cependant, en raison de son coût d'installation élevé, l'utilisation du GPS n'est pas pratique à appliquer pour un grand système de localisation.

Récemment, l'utilisation d'un algorithme inspiré de la nature a attiré l'attention des chercheurs et des praticiens en raison de sa faible complexité et de sa capacité à produire une solution avec des ressources limitées. La plupart des algorithmes inspirés la nature sont des algorithmes stochastiques, qui impliquent plusieurs paramètres aléatoires dans sa procédure. les paramètres permettent aux algorithmes de couvrir simultanément plusieurs zones de la zone de recherche, ce qui augmente les chances de ne pas être piégé dans l'optimum local. [34].

Notre travail consiste à proposer un algorithme de localisation en utilisant une méta-heuristique basé sur l'ordre hiérarchique du poulet et son comportement de groupe, appelé optimisation par l'essaim de poulet (Chicken Swarm Optimization "CSO"), pour minimiser l'erreur de la localisation dans les RCSFs.

Ce mémoire est organisé en quatre chapitres selon le plan méthodologique suivant :

- Dans le premier, Nous présentons un Aperçu général sur les réseaux de capteurs sans fil.
- Le deuxième chapitre présente une étude détaillée sur la Localisation dans les réseaux du capteur sans fil.
- Dans Le troisième chapitre Nous présentons une description pour l'algorithme d'optimisation par l'essaim de poulet (ou CSO).
- Dans le quatrième chapitre, nous présenterons l'implémentation et l'évaluation des performances de notre algorithme

Chapitre I

Généralité sur les réseaux de capteurs sans fil

1 Introduction

Les Réseaux de Capteurs Sans Fil ou (Wireless Sensor Network (WSN) en anglais) est un cas particulier des réseaux ad hoc, il permet de Détecter les paramètres de l'environnement grâce à un nœud capteur (notamment l'humidité, la température, la luminosité).

Ce type du réseau est amené à résoudre divers problèmes comme la surveillance environnementale, les maisons intelligentes, la sécurité, la santé ...etc. Ce chapitre décrit quelques généralités sur les réseaux capteurs sans fil.

2 Définitions

2.1 Un capteur :

Les capteurs sont des objets de taille réduite avec des ressources très limitées, qui sont autonomes et sont capables de traiter les informations de l'environnement qui les entoure et de les transmettre à d'autres dispositifs grâce aux ondes radios sur une distance limitée [1]



Figure I.1 : Exemple d'un Capteur [2]

2.2 Définition d'un RCSF

Un réseau de capteur sans fil est un type spécial de réseau ad hoc avec un grand nombre de nœuds qui sont des micro-capteurs capables de recevoir et de transmettre des données environnementales d'une manière autonome sans interventions humaine.

La position de ces nœuds n'est pas obligatoirement prédéterminée, ils peuvent être aléatoirement dispersés dans une zone géographique appelée « champ de captage » correspondant au terrain d'intérêt pour le phénomène capté (par exemple : lâchée de capteurs sur un volcan pour étudier les phénomènes volcanologique et leur évolution).

Le réseau possède en général un nœud particulier, la base (ou sink), connectée avec les autres nœuds par un réseau filaire ou non filaire est reliée à une alimentation électrique [2]

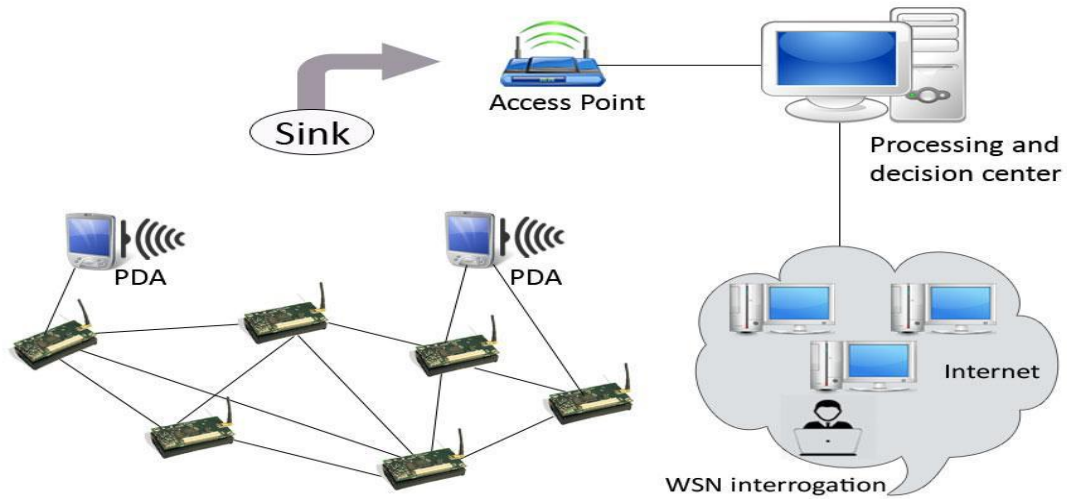


Figure I .2 : réseau de capteur sans fil [3]

3 Architecteur et composant d'un capteur sans fil :

Un capteur est composé Principalement d'une unité de : détection, traitement, stockage, communication, et énergie.

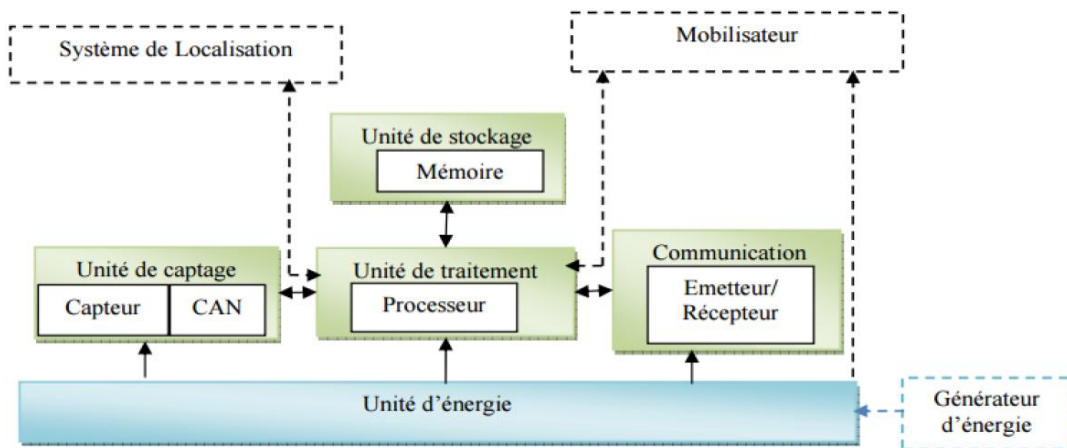


Figure I.3 : architecteur d'un Capteur Sans fil [4]

3.1 L'unité de détection

La fonction principale de l'unité de captage 'détection' est de capturer ou mesurer les données physiques à partir de l'objet cible. Elle est composée de deux sous-unités : le récepteur (reconnaissant la grandeur physique à capter) et le transducteur (convertissant le signal du récepteur en signal électrique). Le capteur fournit des signaux analogiques, basés sur le phénomène observé, au convertisseur Analogique/Numérique (CAN). Ce dernier transforme ces signaux en données numériques et les transmet à l'unité de traitement. Un capteur peut avoir un ou plusieurs unités détection [5]

3.2 L'unité de traitement

Il recueille des données de l'unité de captage ou d'autres capteurs, effectue un traitement sur ces données (si nécessaire) et décide quand et où les envoyer.

Il doit exécuter des programmes et des protocoles de communication différents. Les types de processeurs qui peuvent être utilisés dans un capteur incluent le Microcontrôleur, les DSP (Digital Signal Processors), les FPGA (Field Programmable GateArray) et les ASIC (Application Specific Integrated Circuit) [5]. Parmi toutes ces alternatives, le Microcontrôleur a été le processeur le plus utilisé pour les capteurs à cause de sa flexibilité à être reliés à d'autres composants (comme par exemple l'unité de communication), à son bon prix et sa faible consommation énergétique [5, 6, 7].

3.3 L'unité de communication

Cette unité est responsable de toutes les émissions et réceptions de données via un support de communication sans fil. Les différents choix de média de transmission incluent la Radiofréquence (RF), le Laser et l'Infrarouge [5].

3.4 L'unité énergie

Un capteur est muni d'une source d'énergie, généralement une batterie [8], pour alimenter tous ses composants. Les batteries utilisées sont soit rechargeables ou non. Souvent, dans les environnements sensibles, il est impossible de recharger ou changer une batterie. Pour cela, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs et donc d'un réseau de capteurs.

3.5 L'unité de stockage

L'unité de stockage inclut la mémoire de programme (dont les instructions sont exécutées par le processeur) et la mémoire de données (pour conserver des données fournies par l'unité de captage et d'autres données locales). La taille de cette mémoire est souvent limitée essentiellement par les considérations économiques et s'améliorera aussi probablement au fil des années [9].

4. Type de capteurs :

Il existe actuellement un grand nombre de capteurs, avec des fonctionnalités diverses et variées. La plupart des capteurs dépendent de l'application pour lesquels ils ont été conçus (capteurs aquatiques, sous-terrain, etc. . .). Il est plus intéressant de décrire les capteurs les plus utilisés et leur évolution au cours du temps. La Figure 3 illustre l'évolution des capteurs au cours de ces 20 dernières années. [3]



Figure I.4. Évolution des capteurs [10]

5. L'architecteur de communication d'un RCSF :

L'architecture générale d'un WSN est présentée sur la Figure 4 [3] ; où Chacun des nœuds capteurs a la capacité de collecter des données et les router vers le puits/Gateway et aux utilisateurs finaux.

Les données sont routées au puits via une structure sans fil multi-sauts. Le puits peut, ensuite, communiquer avec l'utilisateur final par Internet ou par satellite ou par n'importe quel type de réseau sans fil (comme le Wifi, les réseaux maillés, les systèmes cellulaires, le WiMax, etc.). Il faut noter qu'il peut y avoir plusieurs puits et plusieurs utilisateurs finaux dans cette architecture. [3]

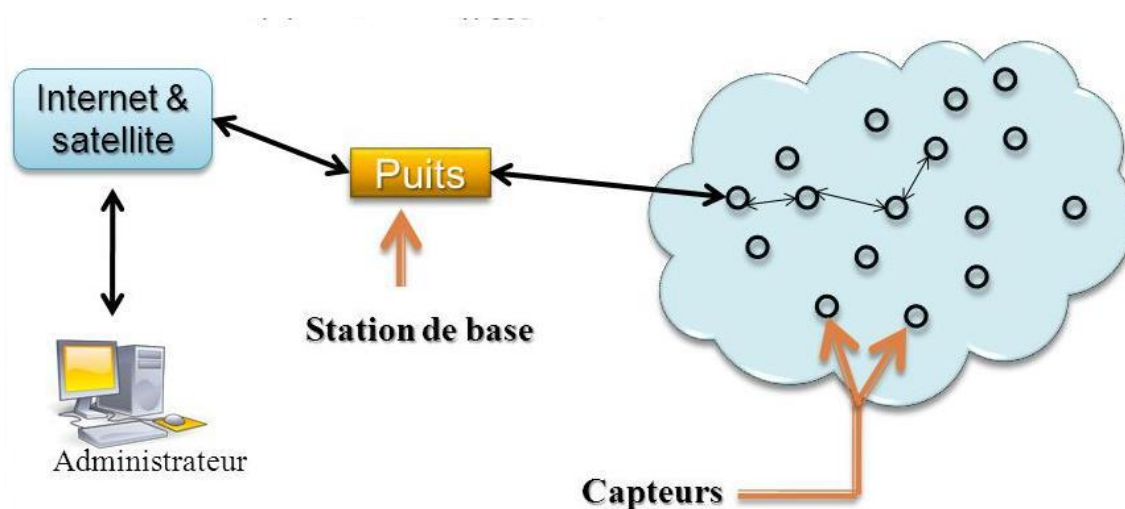


Figure I.5 : Architecture d'un réseau de capteurs sans fil [4]

6. Pile de protocolaire

Un RCSF est une série de connexions entre les capteurs leur permettant de communiquer. Le contenu, la portée, la taille, la vitesse et la fiabilité du réseau dépend d'un ensemble de protocoles et de leur implémentation. Les protocoles sont un moyen de communication prédéterminé. Conceptuellement, il est utile de représenter l'ensemble de ces protocoles sous forme d'une pile, c'est ce qu'on appelle la pile protocolaire. La pile protocolaire utilisée par la station de base, ainsi que tous les autres capteurs du réseau, est illustrée dans la figure 1.5. [12]

Elle comprend cinq couches qui ont les mêmes fonctions que celles du modèle OSI ainsi que trois couches pour de la gestion de la puissance d'énergie, la gestion de la mobilité et la

gestion des tâches. Le but d'un système en couches est de séparer le problème en différentes parties (les couches) selon leur niveau d'abstraction. Chaque couche du modèle communique avec une couche adjacente (celle du dessus ou celle du dessous). Chaque couche utilise ainsi les services des couches inférieures et en fournit à celle de niveau supérieur [11] [13]

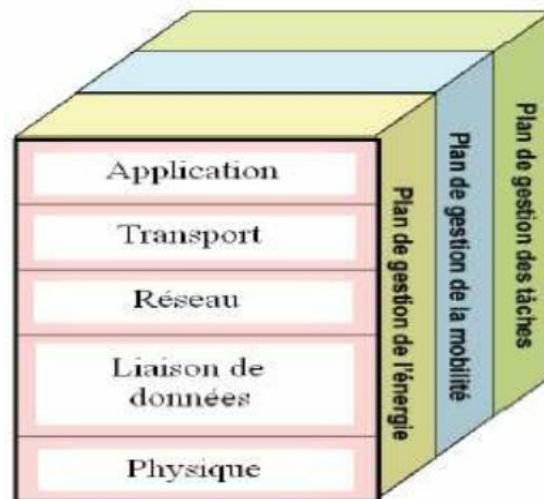


Figure1.6 : la pile protocolaire de communication [14].

6.1 Rôle des couches [15]

- 6.1.1 **La couche physique** : Comme celle du modèle OSI, cette couche est responsable de la modulation, la détection du signal et la sélection des fréquences porteuses.
- 6.1.2 **La couche liaison** : Cette couche est chargée du contrôle d'erreurs, du multiplexage des flux de données, et le contrôle d'accès au média de transmission.
- 6.1.3 **La couche réseau** : L'objectif de cette couche est de trouver des chemins de routage à faible coût d'énergie pour transmettre les données captées vers la station de base. Ainsi, les protocoles de cette couche doivent toujours prendre en compte les limitations en ressources des nœuds capteurs.
- 6.1.4 **La couche transport** : Son rôle est le contrôle du flux, le découpage, l'ordonnancement et le transport des paquets de données, et la gestion des erreurs de transmission.
- 6.1.5 **La couche application** : Afin de fournir une interface d'interaction avec l'utilisateur humain, les nœuds capteurs peuvent être dotés d'une couche application, dont le rôle est d'implémenter l'ensemble d'applications et de logiciels d'interaction

6.2 Plans des gestions [15]

- 6.2.1 Le plan de gestion d'énergie :** Les nœuds capteurs sont sévèrement limités en ressources d'énergie, qui influence directement sur la durée de vie du réseau. Ainsi, le plan de gestion d'énergie doit fournir des mécanismes de gestion efficaces pour réduire le degré de consommation d'énergie, et éliminer les sources de gaspillage de celle-ci.
- 6.2.2 Le plan de gestion de mobilité :** Ce plan est responsable du contrôle du mouvement des nœuds capteurs dans le cas où ils sont mobiles. Il peut par exemple enregistrer les trajectoires d'un nœud capteur afin de l'aider à se localiser.
- 6.2.3 Le plan de gestion de tâche :** Dans un réseau de capteurs, les nœuds peuvent effectuer des tâches qui se diffèrent en termes de consommation de ressources. Ainsi, un plan de gestion de tâche est souvent nécessaire afin de répartir d'une manière équitable les tâches sur les nœuds capteurs, et offrir ainsi une gestion efficace des ressources disponibles

7. Caractéristiques d'un RCSF [4]

- Les réseaux de capteurs sans fil sont caractérisées par :
- Ressources limitées des capteurs en calcul, en mémoire et en énergie.
- Duré de vie limitée.
- Mode de communication directe ou en multi-sauts.
- Densité importante des capteurs qui peuvent attendre des dizaines de millions pour certaines applications.
- Possibilité de découper le réseau en clusters et d'utilise les capteurs comme les calculateurs .
- La coopération entre les nœuds capteurs pour les tâches complexes.
- Deux mode de fonctionnement : « Un à plusieurs » où la station de base diffuse des informations aux différents capteurs, et « plusieurs à un » où les nœuds capteurs diffusent les informations à la station de base .

8. Topologie des RCSF :

Les topologies des réseaux de capteurs sont déterminées à partir des protocoles de routage utilisés pour l'acheminement des données entre les nœuds et le sink. Ces protocoles peuvent être hiérarchiques, plat (Flat).

8.1 Topologie Hiérarchique

Une architecture hiérarchique était proposée pour réduire le coût et la complexité de la plus part des nœuds capteurs en introduisant un ensemble de nœuds capteurs plus coûteux et plus puissant, ceci en créant une infrastructure qui décharge la majorité des nœuds simples a faible coût de plusieurs fonctions du réseau. L'architecture hiérarchique est composée de multiples couches : une couche de capteurs, une couche de transmission et une couche de point d'accès.[16]

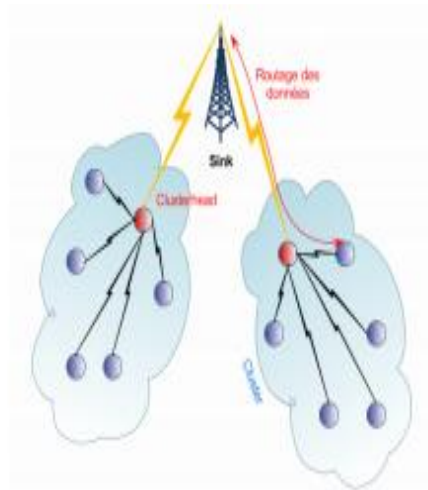


Figure 1.7 : Topologie Hiérarchique [16]

8.2 Topologie plate

Les protocoles à topologie plate (flat) considèrent que tous les nœuds sont égaux, ont les mêmes fonctions, et peuvent communiquer entre eux sans devoir passer par un nœud particulier ou une passerelle. Seul un nœud particulier, le sink, est chargé de la collecte des données issues des différents nœuds capteurs afin de les transmettre vers les centres de traitement. [15]

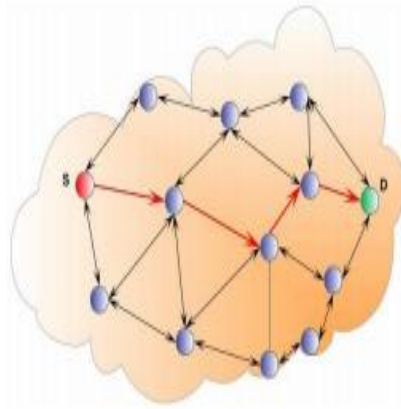


Figure 1.8 : Topologie plate (Flat).[15]

9. Couvertures dans les RCSF

9.1. Couverture :

C'est la surface totale se trouvant en dessous de la marge ou de portée de capteur des données au moins d'un nœud [3].

9.2. Couverture d'un point :

On dit qu'un capteur 'Si' couvre un point 'q' si et seulement si la distance, $d(q, S_i) \leq r_i$, où r_i est le rayon de transmission de capteurs 'Si'. La fonction de distance d peut être la distance euclidienne [3].

9.3. Couverture d'une zone :

On dit qu'un capteur couvre une zone A si et seulement si pour chaque point dans A, la distance $d(q, S_i) \leq r_i$ [3].

10. Connectivité dans les RCSF

La connectivité est un problème majeur dans les réseaux de capteurs (composés d'entités fixes ou mobiles).

On peut considérer un réseau de capteurs sans fil de communication à saut multiple, où tous les nœuds coopèrent dans le but d'assurer des communications entre chacun. Un tel réseau peut être représenté de la manière suivante :

Soit un graphe $G = (V, E)$ représentant le réseau sans fil, avec l'ensemble des nœuds et $E \in V^2$ les arcs donnant les communications directes possibles : (u, v) appartient à E si et seulement si u peut envoyer directement un message à v (on dit alors que v est voisin de u). les couples appartenant à E dépendent de la position des nœuds et de leur portée de communication. Nous prenons l'hypothèse que la portée R de chaque nœud est identique. Soit $d(u, v)$ la distance entre les nœuds u et v . L'ensemble E peut-être défini comme suit:

$$E = \{(u, v) \in V^2 \mid \text{distance}(u, v) \leq R\}$$

Ce graphe est connu sous le nom de graphe disque unitaire, avec R comme rayon de transmission. Dans ce graphe, $G = (V, E)$ nous définissons $n = |V|$ comme le nombre de nœuds dans le réseau. Le voisinage $N(u)$ d'un nœud u représente l'ensemble des nœuds voisins de u , défini par $\{v \mid (u, v) \in E\}$ [3]

11. Domaines D'application des RCSF

Les RCSFs est une nouvelle technologie, qui est actuellement largement répandue dans plusieurs domaines variés ; Ils sont exploités dans le domaine militaire, sécurité civile, médical, transport, environnemental ...etc

11.1 Domaine militaire : Détection et collecte d'information sur la position de l'ennemie et ses mouvements

11.2 Domaine médicale : Suivi les patients à distance (rythme cardiaque, pression du son...), localisation des patients et médecins dans l'hôpital...etc.

11.3 Domaine de transport : Contrôle du trafic, capture de pression des pneus, prévention des accidents...etc. .

11.4 Domaine civil : Apparus dans plusieurs contextes notamment dans la surveillance des habitations (concept de bâtiments intelligents), des infrastructures, des installations et des zones à risques. Leur utilisation permet de réduire considérablement le budget consacré à la sécurité des humains tout en garantissant des résultats sûrs et fiables



Figure I.10 : Quelques domaines d'application des RCSF [8].

12. Contraintes de RCSF

La conception et la réalisation des réseaux de capteurs sans fil sont influencées par plusieurs paramètres [8].

12.1 Durée de vie du réseau :

C'est l'intervalle de temps qui sépare l'instant de déploiement du réseau de l'instant où l'énergie du premier nœud s'épuise. Selon l'application, la durée de vie exigée pour un réseau peut varier entre quelques heures et plusieurs années.

12.2 Ressources limitées :

En plus de l'énergie, les nœuds capteurs ont aussi une capacité de traitement et de mémoire limitée. En effet, les industriels veulent mettre en œuvre des capteurs simples, petits et peu coûteux

12.3 Bande passante limitée :

Afin de minimiser l'énergie consommée lors de transfert de données entre les nœuds, les capteurs opèrent à bas débit. Typiquement, le débit utilisé est de quelques dizaines de Kb/s. Un débit de transmission réduit n'est pas handicapant pour un réseau de capteurs où les fréquences de transmission ne sont pas importantes.

12.4 Facteur d'échelle :

Le nombre de nœuds déployés pour une application peut atteindre des milliers. Dans ce cas, le réseau doit fonctionner avec des densités de capteurs très grandes. Un nombre aussi important de nœuds engendre beaucoup de transmissions inter nodales et nécessite que la station de base soit équipée de mémoire suffisante pour stocker les informations reçues.

12.5 Topologie dynamique :

La topologie des réseaux de capteurs peut changer au cours du temps pour les raisons suivantes :

Les nœuds capteurs peuvent être déployés dans des environnements hostiles (champ de bataille par exemple), la défaillance d'un nœud capteur est, donc très probable.

- Un nœud capteur peut devenir non opérationnel à cause de l'expiration de son énergie.
- Dans certaines applications, les nœuds capteurs et les stations de base sont mobiles.

12.6 Agrégation de donnée :

Dans les réseaux de capteurs, les données produites par les nœuds capteurs voisins sont très corrélées spatialement et temporellement. Ceci peut engendrer la réception par la station de base d'informations redondantes.

13. Conclusion :

Les réseaux de capteurs sans fil sont une réalité fonctionnelle et sont voués à se développer rapidement du fait de la grande variété des domaines d'applications. Cependant, l'utilisation des RCSF reste complexe et posés plusieurs enjeux sont (localisation, routage, alimentation, sécurité,...).

Dans le chapitre suivant, nous présenterons la Localisation dans les réseaux du capteur sans fil.

Chapitre II

Localisation dans les réseaux de capteurs

1 Introduction

La localisation des nœuds dans RCSF est considérée comme l'un des défis inhérents aux réseaux de capteurs sans fil. Les données rassemblées par les capteurs pour de nombreuses applications pourraient n'avoir aucune signification sans reconnaître l'emplacement exact des capteurs. Cette situation déclenche en fait la priorité de la localisation du nœud de capteur sans fil

Dans ce chapitre, nous avons parlé sur le problème de localisation en caractérisant les méthodes

2 Définition

La localisation est un procédé permettant de positionner un objet sur un plan ou une carte géographique, cette opération est réalisée à l'aide d'un terminal capable d'être localiser en temps réel ou de façon différée.

Les positions enregistrées peuvent être stockées au sein du terminal et être extraites postérieurement, ou être transmises en temps réel vers une plateforme logiciel de localisation. La transmission temps réel nécessite un terminal équipé d'un moyen de télécommunication qui permet d'envoyer les positions à des intervalles réguliers [17].

3 Pourquoi la localisation ?

La localisation dans les réseaux de capteurs déployés de manière aléatoire dans la zone de surveillance, en raison soit de l'hostilité de la zone à surveiller, soit de son immensité. Est donc nécessaire pour les raisons suivantes :

- Déterminer les coordonnées géographiques des différents capteurs.
- Localiser les différents événements survenus dans la zone surveillée, mais aussi pour le développement de protocoles de routage de l'information récoltée, pour la couverture de la zone d'intérêt, ... etc
- L'utilisation des capteurs équipés du GPS est trop coûteuse du point de vue financier comme du point de vue énergétique.
- Identifier l'origine de l'information (d'où vient-elle ?) et détecter la place d'un événement est « où se passe-t-il ? »
- Aider à le bon fonctionnement des services des réseaux tels que le routage et les services géo localisation [18] [4]

4. Le processus de la localisation dans les RCSF :

De la problématique de la localisation découle trois problèmes sous-jacents. Les deux premiers sont directement liés au matériel utilisé (définition d'un système de coordonnées et estimation des distances), tandis que le troisième concerne les techniques logicielles utilisées.

- **Définition d'un système de coordonnées (un repère)** : en connaissant les positions de quelques nœuds du réseau (appelés ancres) dans un certain système de coordonnées et les positions relatives des autres nœuds par rapport à ces ancres, il est possible au travers d'un «mapping » de retrouver les positions absolues des nœuds dans le même système. Toute la question demeure de bien « sélectionner » les points repères (les ancres).
- **Estimation des distances** : ce procédé est fortement dépendant du matériel de communication utilisé. En d'autre terme, en collectant des indicateurs de la qualité des communications, les différents nœuds peuvent estimer les distances les séparant les uns des autres.
- **Algorithme de localisation** : les algorithmes de localisation sont utilisés à fin de calculer les positions finales en se basant sur les positions des ancres d'une part et d'autre part sur les estimations inter-nœuds..

D'une manière générale, ce processus consiste à définir un repère de coordonnées, une technique d'estimation des distances inter- nœuds et un algorithme pour la dérivation des positions [19]

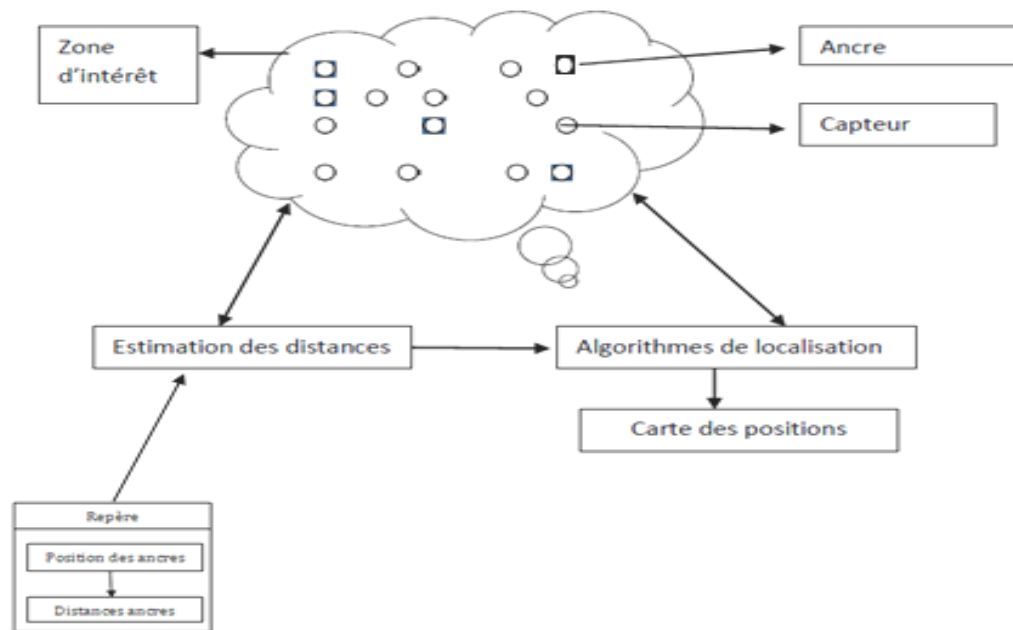


Figure II.1 Processus de localisation dans les réseaux capteurs . [14]

5. Réseaux de capteurs et localisation

5.1 La localisation dans les RCSF

D'une manière générale, presque toutes les méthodes de localisation dans les réseaux capteurs est composée deux phases principales :

5.1.1 Estimation des distances : Cette phase comprend des techniques de mesure pour estimer la distance relative entre les nœuds.

5.1.2 Dérivation des positions : Le but de cette phase est de trouver les positions des nœuds qui respectent au mieux les distances inter nœuds estimées. Si nous connaissons la position de quelques nœuds du réseau dans un certain système de coordonnées, les positions des autres nœuds dans ce système de coordonnées peuvent être trouvées. [17]

5.2 Caractérisation des méthodes :

5.2.1 Utilisation d'estimations de distances :

5.2.1.1 Les méthodes range-free : Ces méthodes ne calculent jamais de distances entre voisins. Elles utilisent d'autres informations telles que la connectivité pour identifier la position des nœuds. Elles semblent donner de bons résultats dans les réseaux denses et réguliers.[20]

5.2.1.2 **Les méthodes range-based** : Ces méthodes estiment les distances entre les nœuds et ensuite dérivent de ces distances les positions des nœuds. [20]

5.2.2 Nécessité de connaître la position d'ancres :

Si une méthode requiert l'encodage au préalable de la position d'un certain nombre d'ancres, cela signifie qu'il faudra qu'une personne intervienne avant un déploiement pour mesurer la position d'un certain nombre de nœuds. Cela est parfois difficile, voire impossible dans certaines situations.

Et donc, le fait que la méthode de localisation requiert ou non de connaître la position d'un certain nombre d'ancres est une caractéristique importante de la méthode. [17]

5.2.2.1 **Les méthodes anchor-based** : Sont celles qui ne fonctionnent pas sans connaître la position d'un certain nombre d'ancres à priori.

5.2.2.2 **Les méthodes anchor-free** : Sont celles qui n'ont besoin de la position d'aucun nœud pour fonctionner ; elles créent donc une carte relative du réseau. Par relative, nous entendons une carte qui est à une translation, une rotation orthogonale, une réflexion et une dilatation près de la 'vraie' carte. Autrement dit, c'est une carte qui conserve les rapports entre les distances entre tous les points.[17]

5.2.3 Forme d'implémentation :

Nous distinguons plusieurs façons d'implémenter le processus de localisation : [20]

5.2.3.1 **Les méthodes centralisées** : Tous les nœuds communiquent avec leurs voisins et renvoient à l'ordinateur central soit des informations sur le signal, soit directement les distances. L'ordinateur central s'occupe si nécessaire d'estimer les distances à partir des informations sur le signal et ensuite de localiser les nœuds.

5.2.3.2 **Les méthodes distribuées** : Ici tous les nœuds communiquent avec leurs voisins pour estimer les distances et échangent leurs informations de voisinage. Ils dérivent ensuite de façon distribuée la position de tous les nœuds dans le réseau. C'est-à-dire qu'à la fin du processus de localisation, chaque nœud doit connaître sa position ainsi que celles de ses voisins et ce sans l'aide d'un ordinateur central qui effectuerait les calculs. Pour les grands réseaux, on considère qu'une méthode distribuée est nécessaire car les méthodes centralisées demanderaient trop de communication pour l'acheminement des informations vers l'unité centrale et consommeraient donc trop d'énergie.

5.3 Critères de localisation : [20]

Voici une liste non exhaustive de critères de localisation. Nous ne pouvons pas tenir compte à tous ces critères au lors de développement d'un algorithme de localisation. Néanmoins, il peut être intéressant de les garder à l'esprit afin de pouvoir rendre notre méthode meilleure selon tel ou tel critère.

- **Précision de la localisation :**

Nous parlons de l'erreur qu'il y a entre les vraies positions des nœuds et les positions calculées par la localisation.

- **Coût énergétique de la localisation :**

Dans les RCSF, une gestion de l'énergie très économique est nécessaire et comme le facteur dominant de la consommation d'énergie est la communication radio, il faut trouver un algorithme qui communique le moins possible via la radio.

- **Robustesse au bruit :**

Il faut analyser comment un algorithme se comporte face au bruit rencontré dans les mesures de distances avec les voisins.

- **Passage à l'échelle :**

Est-ce qu'un algorithme fonctionne sur un réseau de plusieurs milliers de nœuds ? Et si oui, est-il toujours aussi efficace. Ce critère est en rapport avec le fait qu'un algorithme soit implémenté de façon distribuée ou non.

- **Tolérance à la basse connectivité :**

Est-ce qu'un algorithme fonctionne dans un réseau à basse connectivité (un réseau où chaque nœud ne sait communiquer qu'avec un petit nombre de ses voisins) ? Comment sont affectées les performances d'un algorithme face à cette situation ?

- **Réactivité du système :**

Avec quelle rapidité le système de localisation nous renvoie-t-il les positions des nœuds ? Ceci est particulièrement important lorsque l'on veut s'occuper de nœuds mobiles et de suivi de cibles.

6. Estimation des distances

Dans cette partie certaines méthodes importantes pour l'estimation des distances seront étudiées.

7.1 Indicateur de puissance du signal reçu (RSSI).

La mesure de la puissance d'un signal radio reçu ou RSSI est très simple, gratuite et non intrusive. Chaque nœud peut mesurer la puissance du signal reçu pour chaque paquet de données entrant sans impacter ni la bande passante ni l'énergie. Une estimation de la distance entre l'émetteur du paquet et le récepteur est obtenue à partir de cette puissance du signal reçu en connaissant la puissance du signal émis et le modèle d'atténuation utilisé « Pathloss ». Le modèle le plus utilisé est basé sur le log normal shadowing [14]

7.2 Angles d'arrivée (Angle of Arrival (AoA)):

L'angle d'arrivée (AoA) n'est pas basée sur le signal radio seulement, est une technique permettant d'estimer l'AoA du signal de l'ancre par rapport à une référence. Cette méthode consiste à définir une direction entre deux nœuds. La direction (l'angle) est généralement recueillie par la radio et un ensemble de microphones, qui permettent à un nœud écouteur de déterminer sa direction par rapport à l'émetteur et avec des relations géométriques simples on peut calculer les positions des nœuds.[14]

7.3 La différence du temps d'arrivée (TDoA).

La différence entre les moments de réception de plusieurs signaux distincts (TDOA) peut être employée pour estimer la distance entre les nœuds cette distance entre l'émetteur et le récepteur peut être mesuré par la différence de temps d'arrivée avec des différents supports de communication à des vitesses différentes. [14]

7. Dérivation des positions :

La dérivation des positions consiste à calculer les positions finales de chaque nœud capteur en utilisant un des algorithmes de localisation. Chaque algorithme utilise une méthode de calcul qui dépend de la technique d'estimation de distance utilisée.

7.1 La trilatération :

Elle est la méthode la plus simple. Elle est fondée sur le même principe qu'un système GPS : chaque nœud connaissant les positions de trois de ses voisins peut se localiser par l'intersection de trois cercles (utilisant la distance entre le nœud et leur voisin), comme le montre la figure II.2. [14]

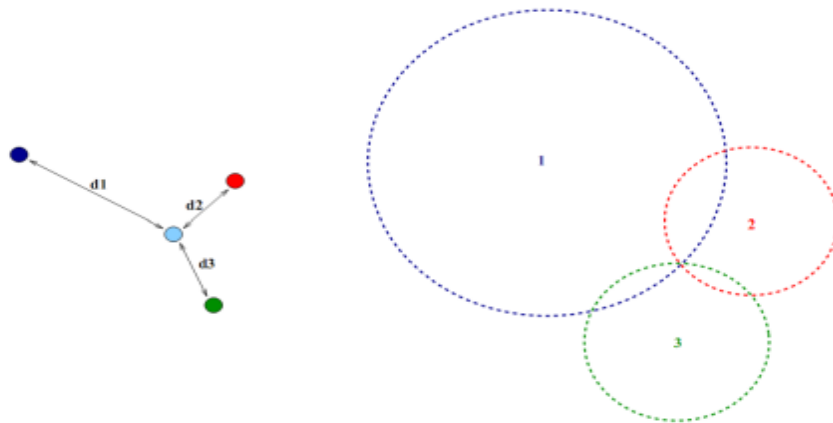


Figure II.2: Graphe représentant la Trilatération [14]

8.2 La triangulation :

La triangulation est une technique permettant de déterminer la position d'un point en mesurant les angles entre ce point et d'autres points de référence dont la position est connue, et ceci plutôt que de mesurer directement la distance entre les points. Ce point peut être considéré comme étant le troisième sommet d'un triangle dont on connaît deux angles et la longueur d'un côté, comme le montre la figure 2.4. [14]

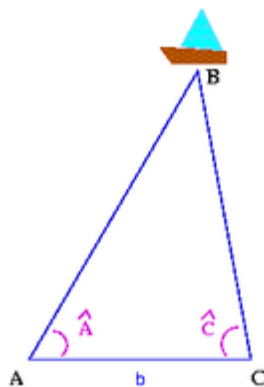


Figure II.3 : Graphe représentant la triangulation [14]

8. Technologie de localisation

On distingue plusieurs types de signaux utilisés dans la localisation selon le type de capteurs, l'environnement ou bien selon la technologie de mesure utilisée.

8.1 Infrarouges

On distingue plusieurs types d'utilisation de l'infrarouge pour la localisation.

Premièrement on peut placer une série d'émetteurs (typiquement des LEDs) infrarouges autour d'une zone et alors un dispositif muni de détecteurs infrarouges peut, en utilisant les caractéristiques des divers signaux infrarouges émis par les différentes sources, se localiser et trouver son orientation dans la zone couverte par les émetteurs [PL04].

On utilise également les infrarouges pour permettre à des robots de se localiser dans un environnement (chaque robot a des émetteurs et des récepteurs infrarouges placés dans plusieurs directions et peut émettre des infrarouges. L'intensité de lumière reçue en retour permet d'estimer la distance entre eux [SR] [3]

8.2 GPS : Global positioning system

La technologie de localisation la plus répandue est sans doute le GPS. Il remplit très bien son rôle de système de localisation à l'échelle planétaire.

Les satellites émettent en continu des signaux radios. Ces signaux radios contiennent une description des trajectoires de chaque satellite. Les récepteurs GPS reçoivent ces informations et connaissent ainsi les positions des satellites. Ils calculent également les temps que prends l'onde radio pour arriver de chaque satellite et en déduisent ainsi les distances jusqu'à chacun d'eux. [3]

8.3 RF-WIFI, RF-Bluetooth :

Les méthodes utilisées pour la localisation dans les réseaux WIFI et Bluetooth sont assez comparables à celles utilisées dans les réseaux de senseurs. Ces deux types de réseaux utilisent des technologies radio [17]

8.4 Image :

Les méthodes de localisation par la vision sont nombreuses, basées sur le traitement d'images et sont très différentes des méthodes de localisation habituelles [17].

8.5. Ultra (sons) :

Il y a eu beaucoup de recherches sur la localisation à base de sons ou ultrasons. Le principal avantage de la technologie est que la vitesse de propagation du son est assez lente, en comparaison à

celle des ondes. Cela permet de mesurer les temps de propagation précisément et ainsi d'obtenir des estimations de distances fiables.

Le système cricket, par exemple, combine des communications par radio et des émetteurs/récepteurs de sons pour se localiser. Pour calculer les distances jusqu'à ses voisins, un 'cricket' envoie simultanément une onde radio et une onde sonore ayant une certaine forme caractéristique. Les nœuds qui reçoivent l'onde radio démarrent un minuteur et attendent l'arrivée de l'onde sonore.

Lorsque celle-ci arrive ils consultent le minuteur et peuvent ainsi déduire le temps de propagation du son. A partir de cela ils calculent les distances de façon assez précise et en déduisent les positions. Pour comprendre cela on peut se souvenir que lorsqu'il y a des orages, on compte parfois le nombre de secondes entre l'éclair lumineux et le coup de tonnerre pour en déduire la distance de l'éclair. Ici l'éclair lumineux est l'onde radio émise et le coup de tonnerre est l'onde sonore caractéristique.

La seule restriction de ce système est qu'il ne peut être utilisé sur de trop longues distances et qu'il est parfois difficile à utiliser dans des environnements trop bruyants ou ayant des obstacles ou des murs. [17]

9. Algorithmes de localisation :

Nous donnons dans cette section un aperçu des méthodes de localisation pour les réseaux de capteurs.

9.1 DV-Euclidean

Cette méthode repose sur la propagation de la distance euclidienne séparant le nœud de l'ancrage.

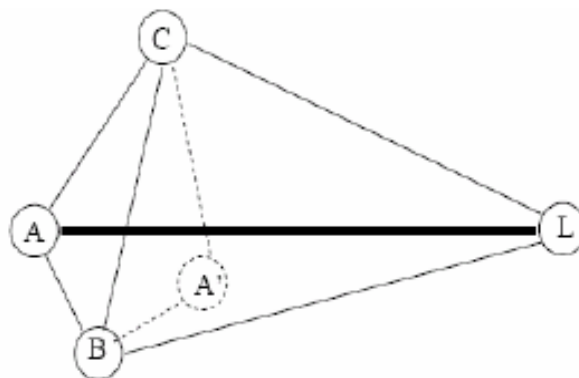


Figure II.4 : Algorithme DV-Euclidean

Soit un nœud A du réseau, pour déterminer sa position, il doit recevoir la position d'au moins deux de ses voisins : B et C. (figure II.4). A doit également estimer les distances AB et AC par mesure de puissance du signal reçu, et BC qui peut être déduite à partir de la reconstitution du voisinage de A. Si on considère le quadrilatère ABCL, où les côtés sont tous connus, la diagonale BC l'est aussi, on peut donc calculer AL qui représente la distance euclidienne de A à l'ancre L. Cette méthode de propagation a l'avantage d'utiliser les mesures réelles, coordonnées GPS par exemple, ce qui évitera la propagation d'erreur de mesure.[17]

9.2 La méthode APIT :

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

Un nœud qui veut se localiser écoute les beacons passants. Pour chaque triplet de beacons différent découvert, il crée un triangle (il connaît les positions des beacons). Il rejette certains d'entre eux via un test appelé PIT test. Il trouve ensuite l'intersection des triangles restants et obtient ainsi une région dans laquelle se trouve le nœud. Le centroïde de cette région est pris comme estimation de position. On peut voir cela de façon intuitive dans la **Figure II.5** [20]

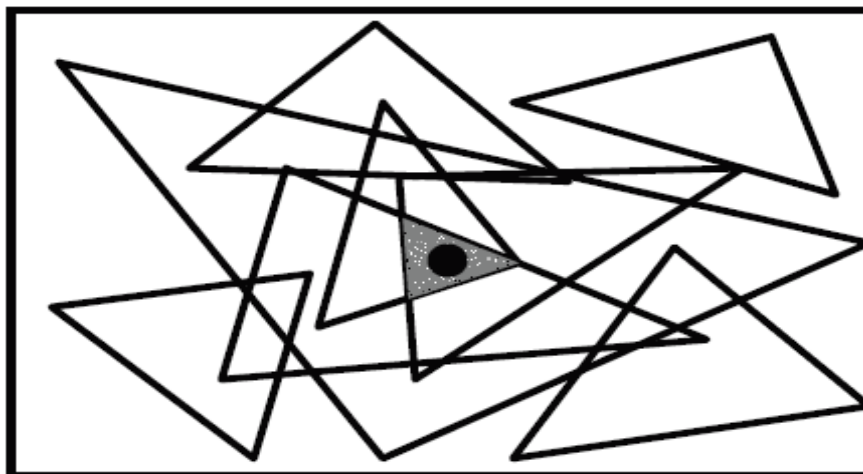


Figure II.5 : La méthode APIT

10. Conclusion :

Il existe dans la littérature plusieurs travaux qui résolvent le problème de localisation dans réseaux de capteurs sans fil, chacun a une méthode pour atteindre ce but.

Nous avons choisi un algorithme méta-heuristique récemment développé appelé optimisation par l'essaim de poulet qui fera l'objet de nos simulations, dans le but de déterminer la position des capteurs.

Donc le chapitre suivant va être expliqué l'algorithme d'optimisation par l'essaim de poulet

Chapitre III

Optimisation par Essaim poulet

1. Introduction :

Les algorithmes d'intelligence en essaim sont une classe d'algorithmes méta heuristiques auxquels on accorde de plus en plus d'attention au cours des dix dernières années. Ces algorithmes imitent le mécanisme d'échange d'informations, de partage d'informations et d'apprentissage parmi les animaux d'essaims en milieu naturel [21]. Le point commun de ces algorithmes est qu'ils obtiennent une solution de bonne qualité après itération.

Optimisation par l'essaim de poulet, ou (chicken swarm optimization (CSO) en anglais), est un nouvel algorithme méta-heuristique proposé par Meng, X.B. Meng, X.B. Et al [22].en 2014 .CSO est un algorithme évolutionnaire qui utilise une population de solutions candidates pour trouve une solution optimale au problème, Il s'inspire du comportement des poules en essaim lors de la recherche de nourriture.

Dans ce chapitre, Nous commençons donc par expliquer Problème d'optimisation, ainsi que définir métras heuristiques Enfin, nous détaillerons le principe de fonctionnement d'un CSO

2. Problème d'optimisation : [23]

Un problème d'optimisation se définit comme la recherche du minimum ou du maximum (L'optimum) d'une fonction donnée. Les variables de cette fonction sont souvent contraintes d'évoluer dans une certaine partie de l'espace de recherche. On aura donc un problème d'optimisation sous contraintes. Mathématiquement, on cherche à minimiser F sur E , c'est à dire, on cherche $x^* \in E$ tel que :

$$f(x^*) = \min(\text{ou max})\{f(x)\}$$

3. Définition Méta-heuristique d'optimisation : [24]

Une méta-heuristique est un algorithme d'optimisation visant à résoudre des problèmes d'optimisation difficile (souvent issus des domaines de la recherche opérationnelle, de l'ingénierie ou de l'intelligence artificielle) pour lesquels on ne connaît pas de méthode classique plus efficace.

Les méta heuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction, par échantillonnage d'une fonction objectif. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution (d'une manière proche des algorithmes d'approximation).

Il existe un grand nombre de méta-heuristique différent, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes utilisent cependant un haut niveau d'abstraction, leur permettant d'être adaptées à une large gamme de problèmes différents.

4.Optimisation par l'essaim de poulet :

4.1 Biologie générale : [25]

Comme la plupart des animaux vivant en groupe, les poules vivent ensemble dans un ordre social très hiérarchisé, ce hiérarchie permet de garder un équilibre cohérent dans le groupe, elle est aussi garante de sa survie. CSO est une méthode stochastique inspirée du comportement des poules en essaim lors de la recherche de nourriture.

4.1.1 L'hiérarchie chez les poules : un comportement inné et naturel : [26]

Trouver sa place dans l' hiérarchie démarre très tôt dans la vie d'une poule ou d'un coq, Ils se picorent, se pourchassent, quelque fois se battent pour la meilleure nourriture ou pour être le plus fort...

L'aspect physique, le plumage, la taille, jouent un rôle important dans la place de chacun dans l'ordre hiérarchique. Mais d'autres facteurs que l'aspect physique entre en jeu :

- L'ancienneté.
- Une forme physique toujours au top.
- Un sujet malade redescend dans la hiérarchie

4.1.2 Les différents comportements hiérarchiques : [26]

En général, les comportements du poulet varient selon le sexe.

Chez les coqs :

- Le coq dominant a la priorité pour l'accès à la nourriture.
- Si un autre coq est présent dans la basse-cour, le dominant le chassera dès qu'il l'approchera de trop près lui et ses groupe. Celui cherchera aussi à lui arracher les plumes du cou et de la queue, symboles de dominance.
- les coqs peuvent appeler leurs compagnons de groupe à manger en premier lorsqu'ils trouvent de la nourriture.
- Le coq combatte les poulets qui envahissent le territoire habité par le groupe.

Chez les poules :

- les poules les plus élevées dans la hiérarchie mangeront et boiront avant les autres.
- Poules plus dominantes qui restent près des coqs de tête.
- Les poulets prépondérants vont dominer les faibles
- chasseront les subalternes des perchoirs S'ils voulaient voler leur nourriture. Leur technique est très simple, quelques petits coups de bec sur la tête ou sur le cou.
- Un comportement aimable existe aussi chez les poules quand elles élèvent leurs enfants (poussin).

Chez les poussins :

- Les poussins à pouvoir se débrouiller seul ou pour trouver sa nourriture. Ont besoin de la becquée des parents pour survivre.
- Les poussins se déplacent autour de leur mère pour aller chercher de la nourriture.



Figure III.1 : comportement poulet dans la nature [27]

4.2 Principe de CSO : [28]

Algorithme CSO développé sur la base de l'ordre hiérarchique et du comportement collectif des poulets. Les algorithmes CSO sont les suivants :

- L'essaim de poules est divisé en plusieurs groupes; chaque groupe composé d'un coq dominant et de nombreux poules et poussins.
- on Basé sur l'identité du poulet et leur fitness, plusieurs groupes seront formés.
- Le poulet suivra les coqs de groupe pendant chercher de la nourriture. Il pourrait y avoir des poulets d'un autre groupe qui essaieront de voler leur nourriture, déjà trouvée par d'autres. L'individu dominant aura d'abord accès à la nourriture.
- Les poussins cherchent de la nourriture autour de leur mère
- Les individus dominants ont un avantage dans la compétition pour la nourriture.
- L'ordre hiérarchique, relation de dominance et la relation mère-enfant dans un groupe sera être inchangé. Tous les poulets vont se mettre à jour leur status après un laps de temps déterminé.

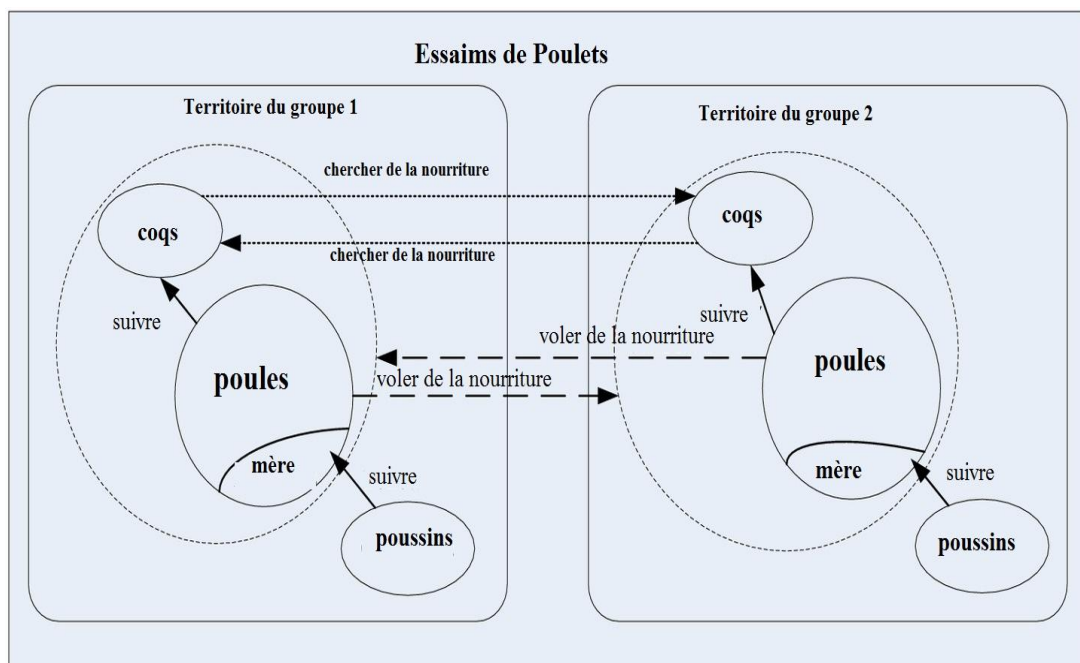


Figure III.2 : Comportement des essaims de poulet. [29]

4.3 Méthodologie de division de groupe : [25]

L'essaim de poulet peut être divisé en plusieurs groupes, chaque groupe contenant : Un dominant coq, beaucoup de poules et beaucoup de poussins. Tout dépend des valeurs de fitness. Les poulets avec le meilleur fitness sont les coqs, Les poulets avec le pire fitness sont des poussins, Les autres seraient les poules

- Le coq sera le chef principal de chaque groupe
- Les poules choisissent au hasard le groupe dans lequel vivre.
- La relation mère-enfant est également établie de manière aléatoire.

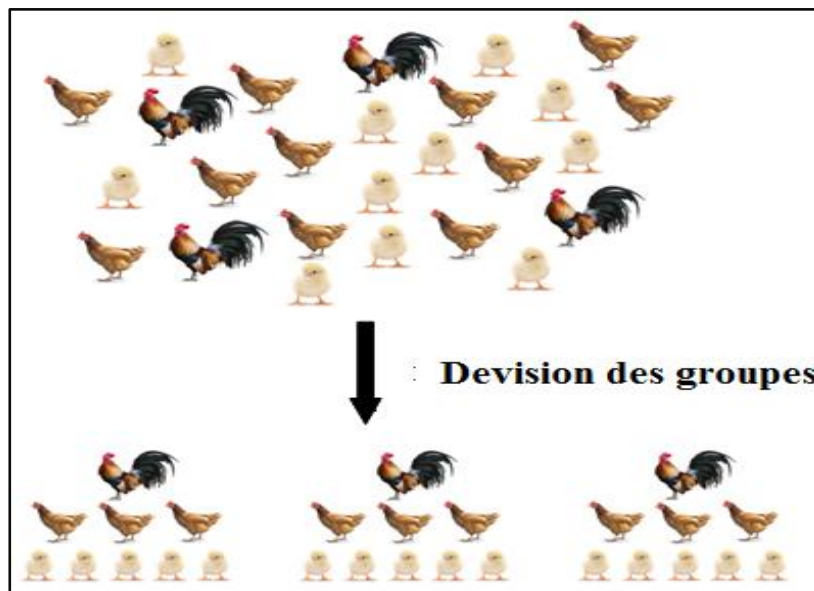


Figure III.3 : système hiérarchique de l'essaim de poules

4.4 Mouvement des poulets : [30]

Tous les N poulets virtuels, représentés par leurs positions $x_{i,j}^t$ ($i \in [1 \dots N]$, $j \in [1 \dots D]$) au pas de temps t , recherchent de la nourriture dans un espace D dimensionnel. Dans ce travail, les problèmes d'optimisation sont les minimums.

Ainsi, les meilleurs poulets \rightarrow ceux avec des valeurs de fitness minimales.

4.4.1 Mouvement des coqs :

Les coqs avec de meilleures fitness ont la priorité pour l'accès à la nourriture que ceux qui ont de moins valeurs de fitness. Pour plus de simplicité, les coqs avec de meilleures fitness peuvent rechercher de la nourriture dans un plus grand nombre de lieux que celui des coqs avec des valeurs de fitness pires. Les mouvements des coqs sont :

$$x_{i,j}^{t+1} = x_{i,j}^t * (1 + \text{randn}(0, \sigma^2)) \quad (\text{III} - 1)$$

- $\text{randn}(0, \sigma^2)$ est une distribution gaussienne avec moyenne 0 et déviation

$$\sigma^2 = \left\{ \begin{array}{l} 1 \text{ si } f_i \leq f_k \\ e^{\frac{(f_k - f_i)}{|f_i| + \varepsilon}} \end{array} \right\} k \in [1, N] k \neq i \quad (\text{III} - 2)$$

- ε , qui est utilisé pour éviter une erreur de division par zéro, est la plus petite constante de l'ordinateur
- k , un index de coq, est sélectionné de manière aléatoire dans le groupe des coqs
- f est la valeur de fitness du x correspondant.

4.4.2 Mouvement des poulets

Les poules peuvent suivre leurs coqs de groupe pour chercher de la nourriture. Ils volaient la bonne nourriture trouvée par d'autres poulets Bien qu'ils seraient réprimés par les autres poulets. Les poules les plus dominantes auraient avantage à concurrencer pour la nourriture que les moins dominantes

$$x_{i,j}^{t+1} = x_{i,j}^t + S1 * \text{rand} * (x_{r1,j}^t - x_{i,j}^t) + S2 * \text{rand} * (x_{r2,j}^t - x_{i,j}^t) \quad (\text{III} - 3)$$

$$S1 = e^{\frac{(f_k - f_r1)}{|f_i| + \varepsilon}} \quad (\text{III} - 4)$$

$$S2 = e^{(f_{r2} - f_i)} \quad (\text{III} - 5)$$

- Rand est un nombre aléatoire uniforme sur [0, 1]
- $r1 \in [1, \dots, N]$ indice du coq, qui est le compagnon de groupe du poule
- $r2 \in [1, \dots, N]$ indice du poulet (coq ou poule), choisi au aléatoire $r1 \neq r2$

4.4.3 Mouvement des poussins

Les poussins se déplacent autour de leur mère pour chercher de la nourriture. Ceci est formulé ci-dessous

$$x_{i,j}^{t+1} = x_{i,j}^t + FL * (x_{m,j}^t - x_{i,j}^t) \quad (\text{III} - 6)$$

- $x_{m,j}^t$ Représente la position de l'item la mère du poussin ($m \in [1, N]$)
- FL ($FL \in (0, 2)$) Le paramètre indique que le poussin suivrait sa mère pour chercher de la nourriture considérez les différences individuelles,
- Le FL de chaque poussin choisirait à l'aléatoire entre 0 et 2

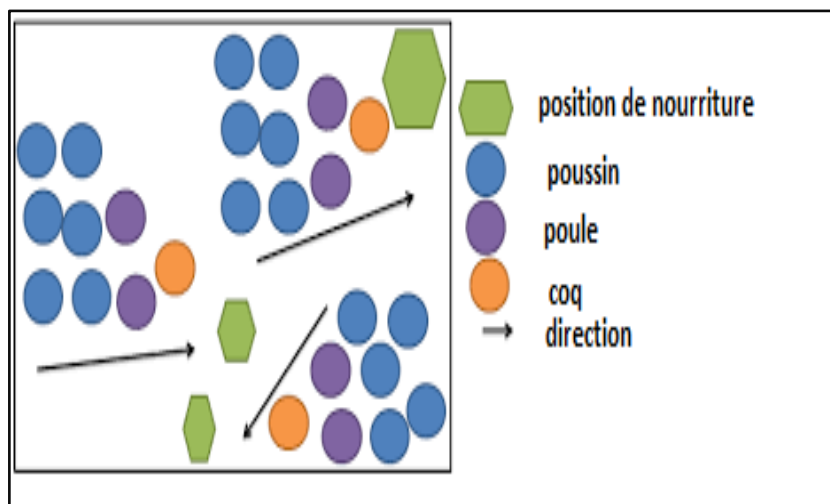


Figure III.4 : mouvement poules en essime. [31]

4.5 Les étapes CSO : [25]

Les étapes du CSO sont :

- **Étape 1** : initialiser la Population, et définis les paramètres puis calcul la valeur de fitness des chacun individus.
- **Étape 2** : Triez les individus par valeur fitness et déterminez les identités et les sous-groupes.
- **Étape 3** : Mettre a jour les positions et les valeurs de fitness des poulets avec des identités différentes selon Eq(III -1) Eq(III - 3) Eq(III - 6) .
- **Étape 4** : sélectionne la solution optimale parmi la population

4.6 Algorithme CSO :

- 1) Initialise le paramètre de CSO (population totale **NP**, le nombre coqs **Nr**, le nombre poules **Nh**, le nombre poussins **Nc** et le nombre maximum de générations **itéramax**)
- 2) Générer une population $x = (x_1, x_2, \dots, x_i, \dots, x_{NP})$ de poulets de taille **Np** réelle dans des positions aléatoires.
- 3) Calculer les fitness **f** (x_i) et trouver la meilleure x_{best} position de chaque individu de la population.
- 4) **for** itéra = 1 : **itéramax**
- 5) trie les individus de la population en fonction de fitness .
- 6) Déterminer les identités, diviser l'essaim en différents groupes et déterminer la relation entre les poussins et leur mère dans un groupe.
 % Mouvement des poulets
- 7) **for** i = 1 : **NP** if i = nombre coq Mettez à jour son emplacement à l'aide de l'équation (III 1) et Calcule nouvelle valeur de fitness ; end if **end for**
- 8) **for** i = 1 : **NP** if i = nombre poulet Mettez à jour son emplacement à l'aide de l'équation (III 3) et Calcule nouvelle valeur de fitness ; end if **end for**
- 9) **for** i = 1 : **NP** if i = nombre poussins Mettez à jour son emplacement à l'aide de l'équation (III 6) et Calcule nouvelle valeur de fitness ; end if **end for**
- 10) Calcule nouvelle valeur de fitness **Fn** (x_i).
 %Fonction de traitement transfrontalier améliorée Évaluer les valeurs de fitness
- 11) Si la nouvelle valeur de fitness est meilleure que la valeur de fitness de l'individu actuel mettre à jour la position et la valeur fitness de l'individu.
- 12) Si la nouvelle valeur de fitness est supérieure à la valeur de fitness globale actuelle du meilleur individu, mettez à jour la position et la valeur de fitness du meilleur individu global actuel.
- 13) **end for**
- 14) Si un critère d'arrêt est rempli, indiquez la position et la valeur de fitness du meilleur individu global à la population

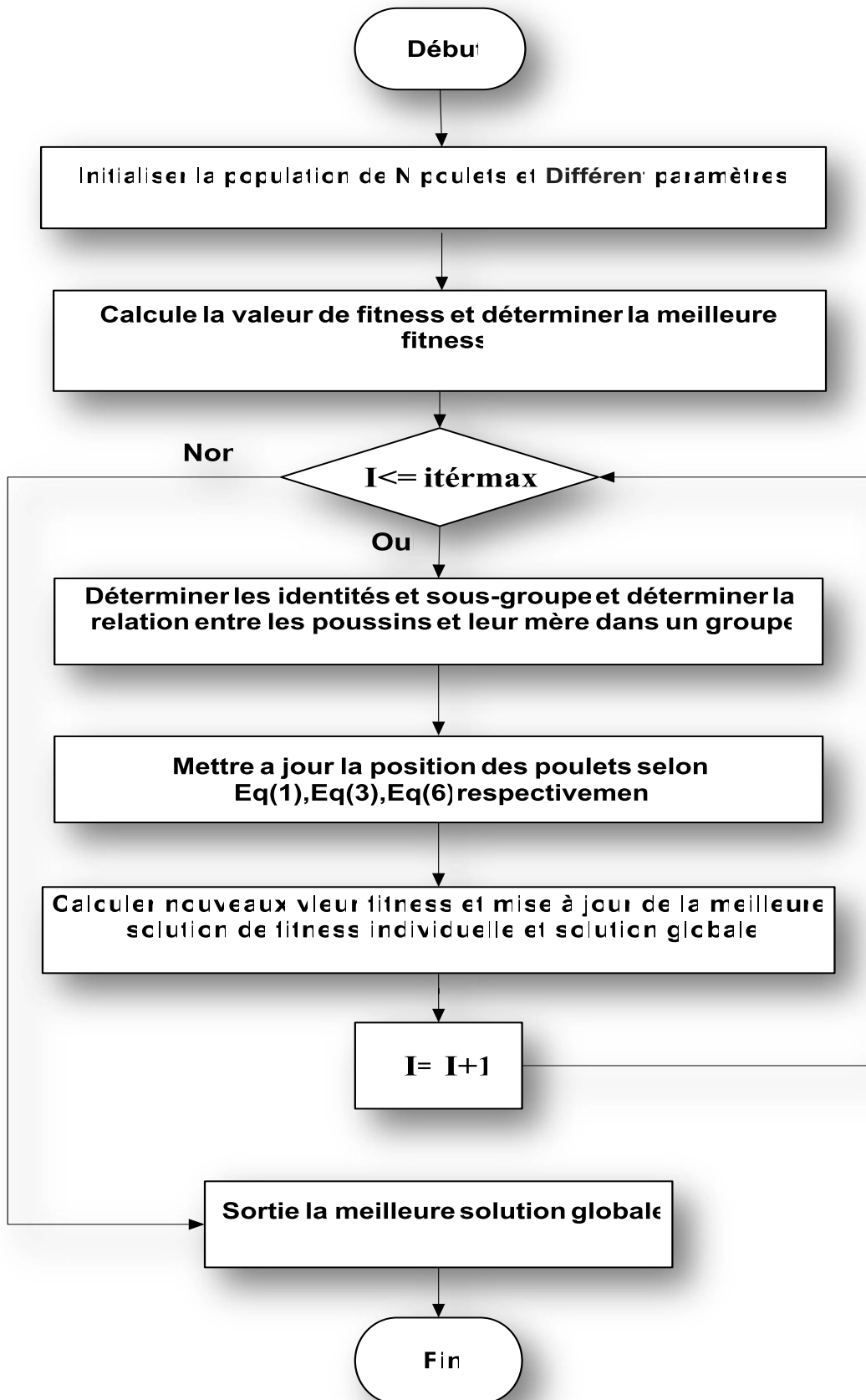


Figure III.4 : L'organigramme de l'algorithme CSO

5. Conclusion :

Au cours des dernières années, diverses méthodes d'optimisation heuristique ont été développées. Dans ce chapitre, nous avons présentons un algorithme d'optimisation appelé L'optimisation par l'essaim de poulets à pour objectif de trouver les régions optimales de l'espace de recherche complexe grâce à l'interaction des individus de la population.

Dans le dernier chapitre nous présentons notre implémentons et évaluation de cette Algorithme d'optimisation dans La localisation des capteurs.

Chapitre IV

Implémentation et Discussion

1. Introduction :

Dans ce dernier chapitre, on va présenter notre contribution sur l'application de l'algorithme d'optimisation par essaim de poulet pour optimiser la précision de la localisation des nœuds dans les RCSFs. Pour mettre en œuvre notre proposition nous avons utilisé le langage de simulation MATLAB, et afin de juger la performance de l'algorithme CSO dans la localisation leurs résultats sont comparés avec ceux de la méta-heuristique PSO.

2. Le langage de programmation utilisé :

Nous avons choisi MATLAB (MATrix LABoratory), comme un environnement d'implémentation pour notre application. MATLAB est un langage de calcul scientifique de haut niveau et un environnement interactif pour le développement des applications, la visualisation et l'analyse de données, ou encore le calcul numérique.

Nous utilisons MATLAB® 2013 pour résoudre des problèmes de calcul scientifique plus rapidement. Le langage MATLAB met à la disposition du développeur les opérations vectorielles et matricielles, fondamentales pour les problèmes d'ingénierie et scientifiques.

Il permet un développement et une exécution rapide à l'égard du langage MATLAB nous pouvons programmer et tester des algorithmes plus rapidement qu'avec les langages traditionnels, car il n'est pas nécessaire d'effectuer les tâches de programmation de bas niveau, comme la déclaration des variables, la spécification des types de données et l'allocation de la mémoire.

Nous avons développé notre application à l'aide du langage MATLAB version R2013a sur Windows 8 Professional 32 bit, de RAM de 2.00 Go, et de processeur Intel(R) Core(TM) i3-2310 M CPU @ 2.10 GHz, 2.10GHz.

L'environnement de MATLAB possède 4 fenêtres

- 1 Au centre l'invite de commande (command window).
- 2 En haut à droite le contenu de l'espace courant de travail (workspace).
- 3 gauche la liste des fichiers du répertoire courant (currentfolder).
- 4 En bas à droite l'historique des commandes tapées (command history)
- 5 Fenêtre GUI en MATLAB

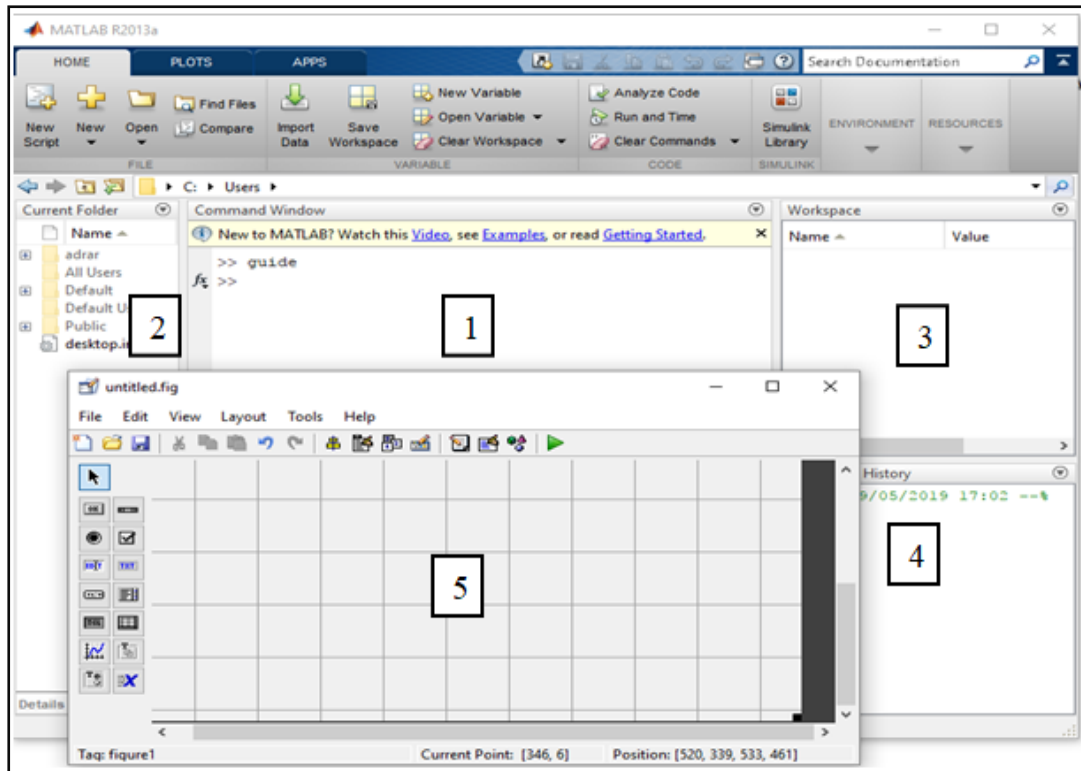


Figure IV.1.L'interface du MATLAB.

3. Description générale de l'algorithme proposé

Afin d'améliorer la précision de localisation, on a utilisé un nouvel algorithme méta-heuristique basé sur l'optimisation par l'essaim de poulets (CSO) et les ancrés a été mis en place dans RCSF.

a) Formulation de problème :

Le présent travail de recherche a pour but de localiser N capteurs déployés aléatoirement dans un plan bidimensionnel de taille $L \times L$ représente la zone d'intérêt de RCSF ; Pour ce la nous avons considéré $m \geq 0$ nœuds appelés ancrés dont ces positions sont connue a priori, donc l'objectif de la procédure de localisation est de trouver les positions de n nœuds inconnu étiquetés $m+1, \dots, N$ ($N=n+m$). Chaque nœuds de réseaux sont caractérisé par un rayon de transmission \otimes similaire.

L'estimation de position des nœuds cibles peut être formulée comme un problème d'optimisation, impliquant la minimisation d'une fitness représentant l'erreur dans la localisation des nœuds cibles. La somme des erreurs de distance au carré entre les nœuds cibles et les ancrés voisins peut être considérée comme la fonction objective de ce problème [32]

Soit (x, y) les coordonnées du nœud cible et d_i soit la distance entre le nœud cible et le j^{ieme} ancre. La distance est calculée en utilisant la formule suivante :

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (IV.1)$$

Soit d_i la valeur de d_i obtenue à partir des mesures de distance. La fonction objective du problème de localisation définie comme suite :

$$f(x, y) = \frac{1}{M} \sum_{j=1}^M |\sqrt{(x' - x_j)^2 + (y' - y_j)^2} - d_j| \quad (IV.2)$$

- ✓ $M > 3$ nombre des nœuds des ancres dans la zone de transmission du nœud cible
- ✓ (x', y') coordonne du nœud à estimer.
- ✓ (x_j, y_j) coordonne du nœud ancre j^{ieme} .
- ✓ d_j distance réel entre capteur inconnu et ancre.

b) algorithme de localisation proposé :

1. Identification de la population initiale :

déployer aléatoirement dans la zone d'intérêt N_p (nombre totale des individus) poulets (en préciser N_r nombre de coq, N_h nombre de poulet et N_c nombre de poussin).

2. calculer la valeur de fitness :

Calculer la valeur de fitness de chaque individu dans population selon l'équation (IV 2)

Détermine meilleures valeurs de fitness représente valeur minimale f_{best} . et la position du meilleur individu (x_{best}, y_{best}) .

3. Déterminer identité et sous-groupe de poulet :

Dans cette étape on va Identifier l'identité des poulets (coq, poule et poussin), trier tous les individus de la population selon fitness, les poulets N_r avec les valeurs de fitness minimales sont désignés comme des coqs, tandis que les poules du N_c avec les valeurs de fitness maximal seraient considérés comme des poussins et les restes sont des poules.

L'essaim est divisé en différents groupes, Chaque coq représente un groupe et considère comme chef principal de groupe, les poules choisissent au hasard le groupe dans lequel vivre et les poussins choisissent aléatoire des poules comme mères et rejoignent les groupes correspondants.

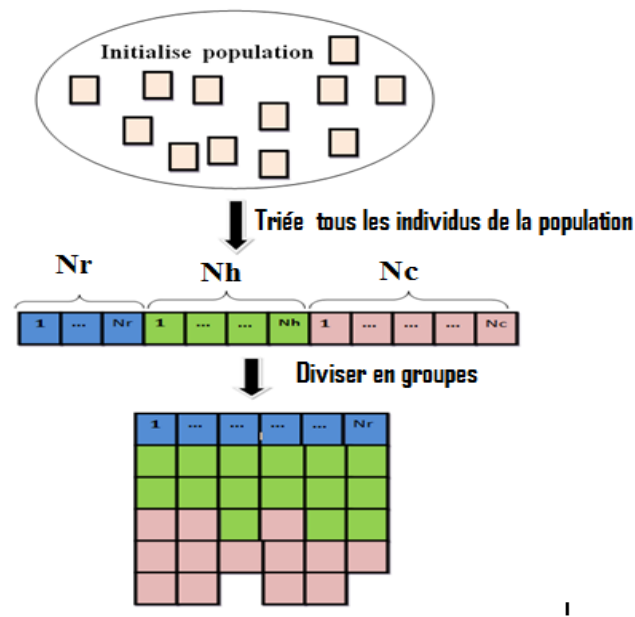


Figure IV.2. Déterminer les identités et sous-groupes

4. mouvement des poulets :

Calcule a l'aide des l'équation (III - 1), (III - 3) et (III - 6) respectivement (Voir le chapitre III) les nouvelle coordonnées de chaque individu des poulets selon leur les identités différentes

NB : Si les nouvelles coordonnées déplacent l'espace de recherche $L \times L$, alors recalculez ces positions.

Calcule nouvelle valeur de fitness, Si la nouvelle valeur de fitness est meilleure que la valeur de fitness de l'individu actuel mettre à jour la position et la valeur fitness de l'individu. Sinon garder la même position.

5. mettre à jour la solution globale :

Si un des nouvelles valeurs de fitness inférieur a la valeur de fitness globale f_{best} , mettre à jour la position (x_{best}, y_{best}) et la meilleur valeur de fitness f_{best} .

6. fournir la meilleure solution globale :

Répétez les étapes 3,4 et5 jusqu'à atteindre le nombre maximal d'itérations, puis la position finale du capteur inconnu.

c) CSO amélioré :

L'algorithme CSO peut estimer l'emplacement de capteurs inconnus uniquement si le nombre d'ancres dans la plage de transmission est supérieur ou égal à 3.

On a ajouté des améliorations à l'algorithme CSO Pour nous pouvons localiser le plus grand nombre de capteurs inconnus. Chaque capteur connaissant suffisamment 'ancres dans son voisinage, devenir une ancre estimée Seulement si l'erreur entre sa position estimée et sa position réelle est inférieure à seuil ε (valeur fitness inférieur à ε). Cette information est alors diffusée dans le voisinage an de permettre aux autres capteurs de se localiser.

4. L'algorithme propose :

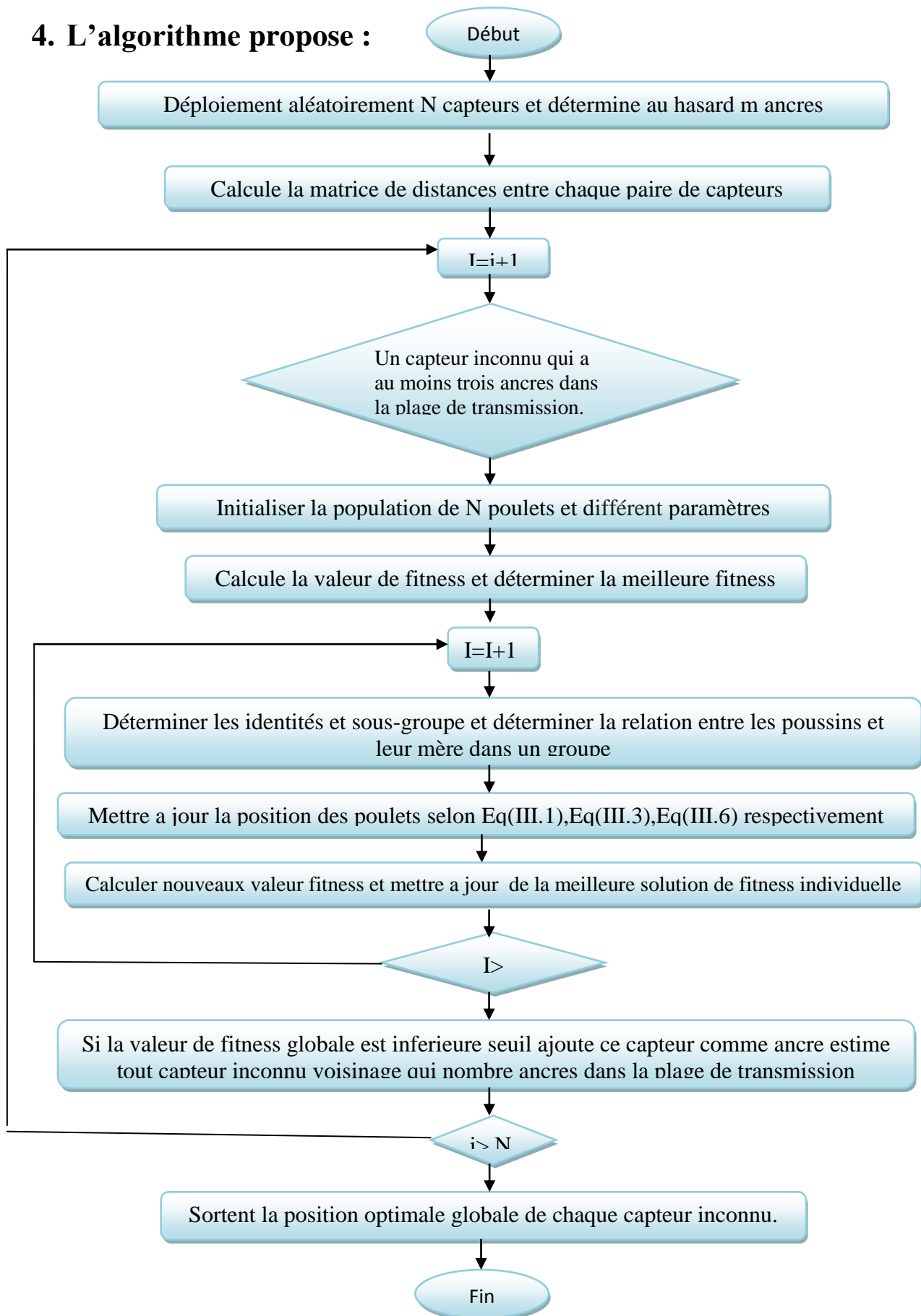


Figure IV.3. L'organigramme propose.

5. Les étapes de la simulation :

Pour exécuter notre application suivre les étapes suivantes :

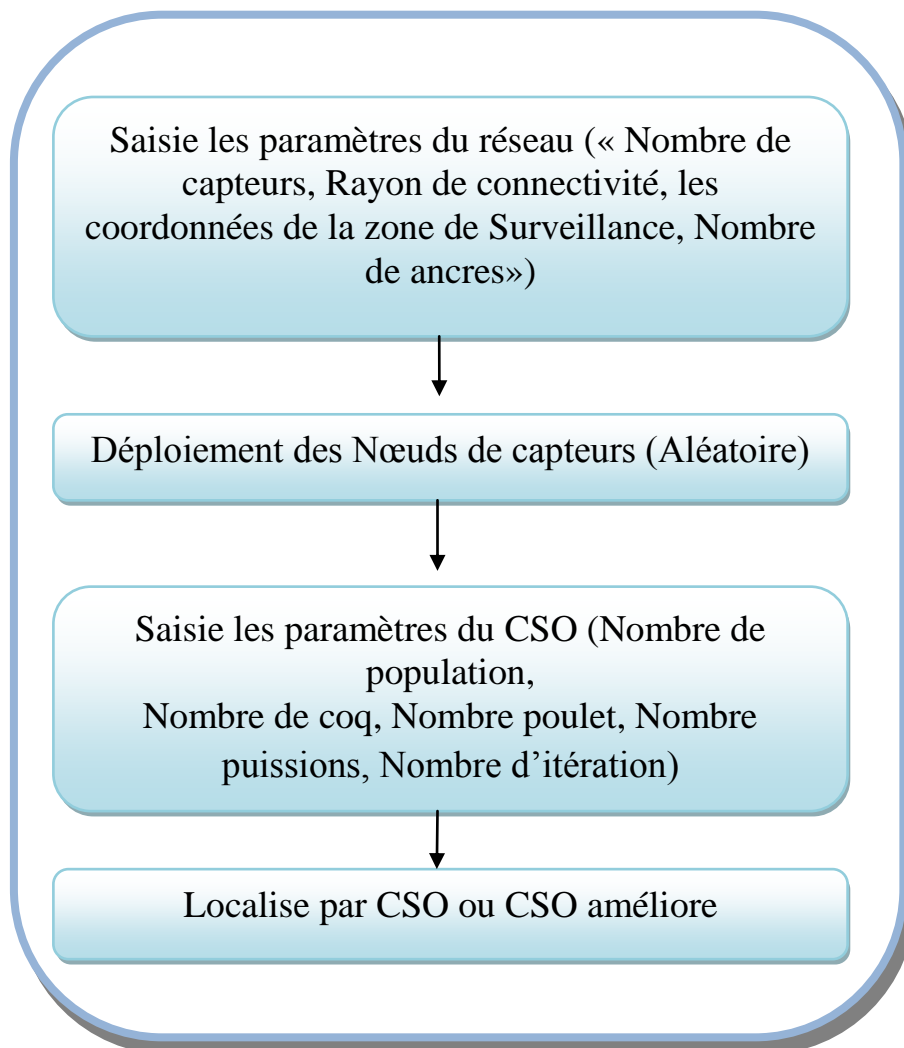


Figure IV.4 : Les étapes de simulation

6. Description des étapes d'exécution d'application :

L'interface principale de notre application est présentée dans la figure. IV.5

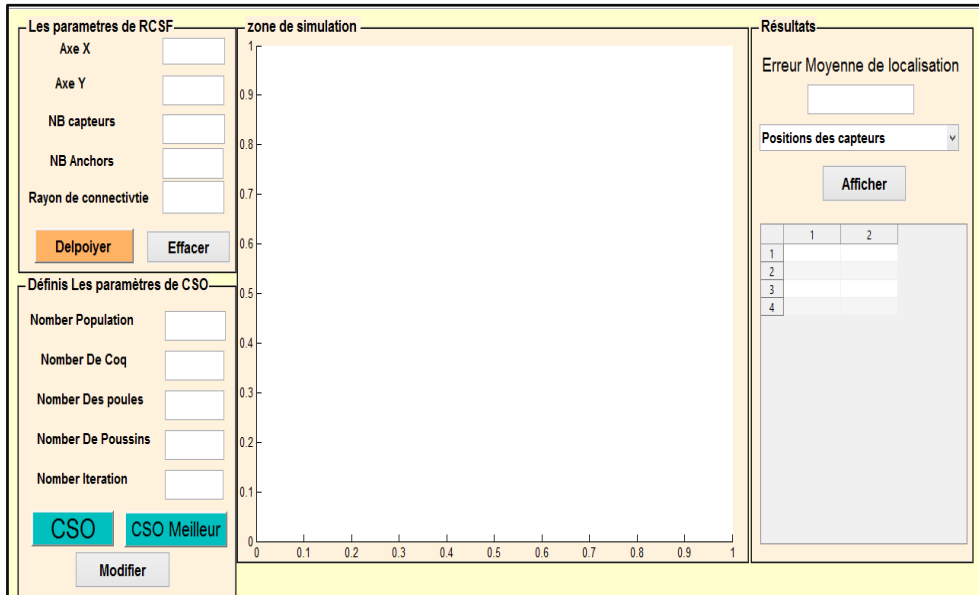


Figure. IV.5 : L'interface de simulation.

6.1 Choix des paramètres de réseau : Pour lancer l'algorithme de CSO, il faut d'abord saisir tous les paramètres initiaux suivant

- ✓ Nombre des capteurs :
- ✓ Nombre des ancrs
- ✓ Le rayon de connectivite
- ✓ La zone de surveillance

6.2 Déploiement des capteurs : Après de saisir les paramètres de réseau, on a déployé aléatoirement les capteurs dans la zone de simulation par : un click sur le bouton « déployer »

6.3 Choix du paramètre du CSO : Après de saisir tout les paramètres de initiaux de réseau, on doit saisir les paramètres du l'algorithme CSO définis, à savoir :

- ✓ Nombre de population.
- ✓ Nombre de coq.
- ✓ Nombre poulet.
- ✓ Nombre puissions.
- ✓ Nombre d'itération.

6.4 Localiser par CSO : Deux approches de localisation par l'algorithme CSO ont proposées dans cette application à savoir :

- ✓ **CSO** : On applique l'algorithme CSO par cliquant sur le bouton « CSO »

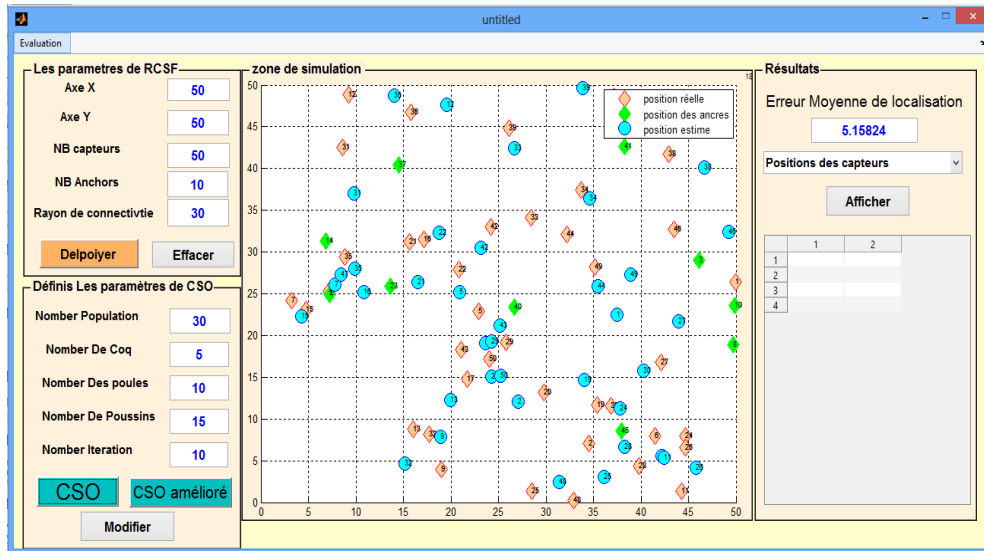


Figure. IV.6 : La localisation avec CSO

Les losangées oranges représentent les capteurs déployés aléatoirement, les losangés vers représentent les ancre et Les cercles bleus indiquent les capteurs localisés par CSO.

- ✓ **CSO amélioré** : Pour applique l'algorithme CSO amélioré en cliquant sur le bouton « CSO amélioré »

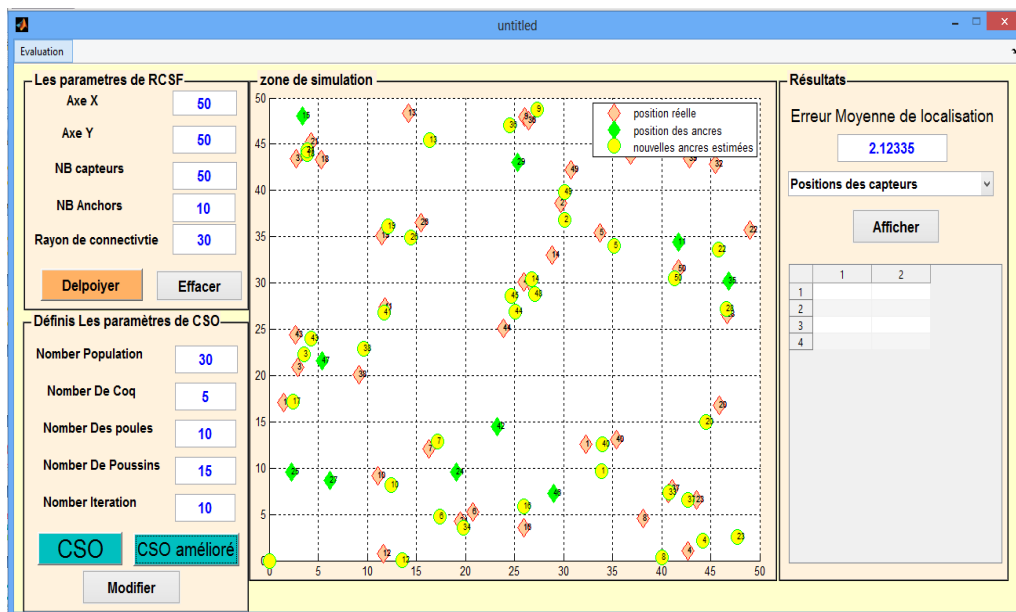


Figure. IV.7 : La localisation avec CSO amélioré.

Les cercles jaunes indiquent les capteurs considérés comme ancre estimée. On remarque que l'erreur de localisation diminue aussi les coordonnées estimées par CSO amélioré sont plus proches des coordonnées réelles que celles estimées par CSO classique, et on peut localiser plus nombre de capteur inconnu.

6.5 L'Affichage des informations supplémentaire :

Pour afficher les informations en cliquant sur le bouton « Affichage », Ces informations sont :

- a. Positions des capteurs : les coordonnées réelles de nœud de capteur
- b. Positions des ancres : les coordonnées des ancres.
- c. Distance entre aux d capteurs : la matrice de distance entre les capteurs $N \times N$
- d. Distance entre les capteurs et les ancres : la matrice de distance entre les capteurs et les ancres $N \times m$
- e. N° Anchor lié par au capteur : les nombres des ancres dans la zone de transmission.
- f. Identité et sous-groupe : déterminer type de poulet selon les valeurs de fitness (coq, poulet ou bien poussins), et déterminer les sous groupe selon le nombre des coqs
- g. Best Fitness : La meilleure valeur optimale de fitness
- h. la Meilleure position : les coordonnées estimées par l'algorithme CSO amélioré

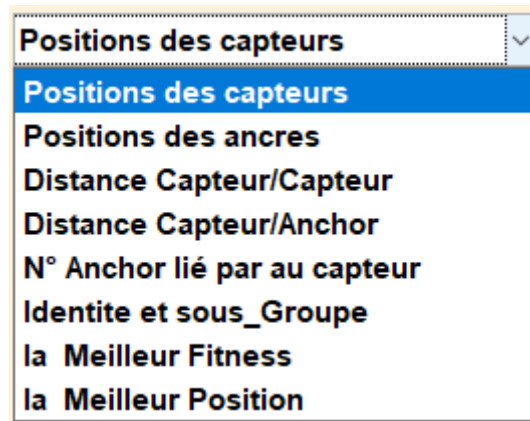


Figure IV.8 : L'information supplémentaire

7. Évaluation l'algorithme de CSO :

Le but typique de cette application est d'évaluer les résultats obtenus par l'application de l'algorithme CSO. Dans notre simulation, nous intéresserons essentiellement à la précision de

localisation de l'algorithme CSO comme un critère d'évaluation puisqu'elle constitue une phase critique très importante qui permet de montrer l'efficacité de cette méthode de localisation.

7.1 La précision de l'algorithme

Afin d'évaluer la précision dans l'algorithme on peut calculer l'erreur moyenne produite par cette méthode. On doit calculer l'erreur moyenne comme étant la moyenne des différences entre les positions réelles et les positions estimées. L'erreur moyenne est calculée comme suit :

$$\text{Erreur} = \frac{\sum \sqrt{(X_r - X_c)^2 + (Y_r - Y_c)^2}}{N}$$

Tel que X_r, Y_r : les positions réelles d'un nœud

X_c, Y_c : les positions d'un nœud trouvées par l'algorithme proposé.

N : nombre de capteurs

7.2 Évaluation de notre application

Pour savoir l'effet des différentes paramètres sur la précision de l'algorithme CSO, on a effectué la série d'expériences suivante :

Le tableau ci-dessus représente les paramètres fixés pour l'évaluation :

Les paramètres	Les valeurs
zone de surveillance	100*100m
Nombre des capteurs	60
Nombre des ancres	20
Rayon de connectivité	40
La taille de population	40
Nombre d'itérations	50

Tableau IV.1 : Les paramètres d'évaluation

a. La précision par rapport au nombre de capteurs :

Dans cette expérience on fixe les différents paramètres sur les valeurs dans le tableau IV.1 et on varie le nombre de capteurs.

Les résultats obtenus sont présentés sur la figure **IV.9**

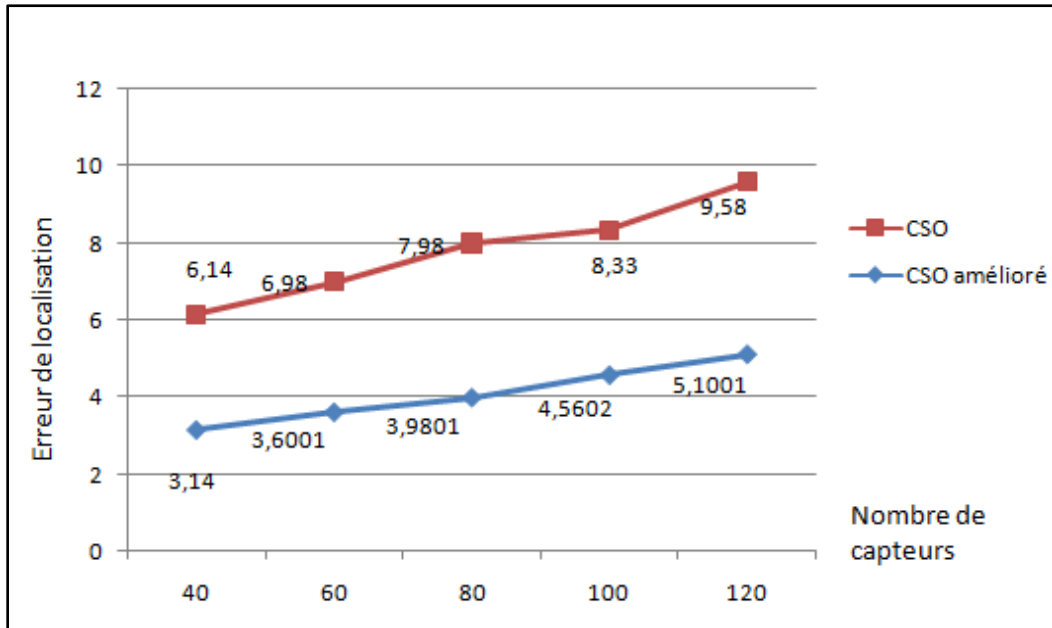


Figure IV.9 : Erreur de localisation Vs nombre de capteurs.

Cette figure IV.9 montre clairement que la valeur de l’erreur moyenne est augmentée avec l’augmentation de nombre de capteurs dans les deux approches de CSO et on peut savoir aussi que les valeurs d’erreur moyenne de CSO amélioré est meilleure que CSO.

b. Précision par rapport au rayon de connectivite

De la même manière, l’erreur moyenne par rapport le rayon de couverture sera calculé.

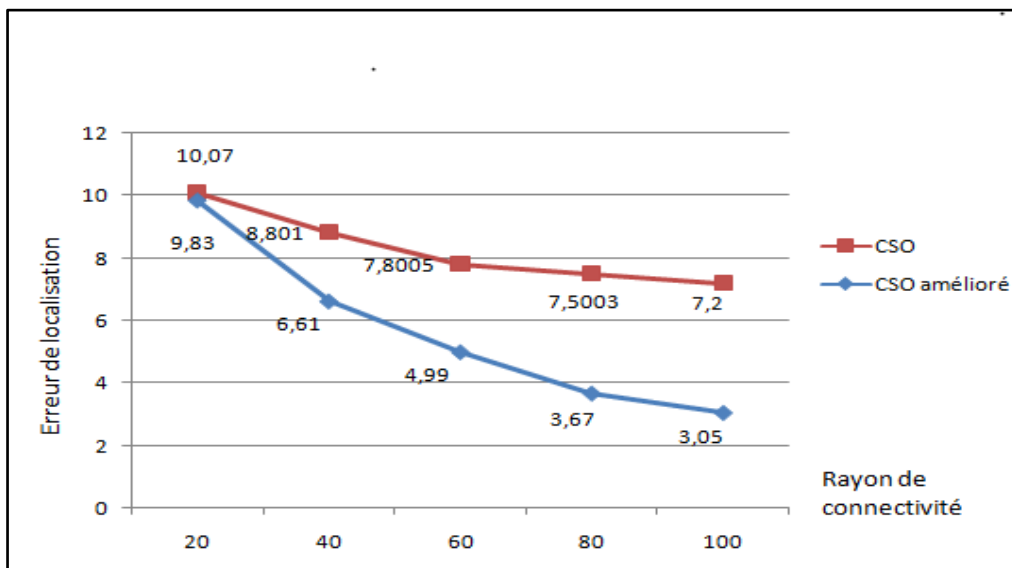


Figure IV.10 : Erreur de localisation Vs rayon de connectivite

D'après la figure IV.10, on voit que l'erreur moyenne diminue avec l'augmentation de rayon de connectivité pour les deux algorithmes : CSO et CSO amélioré.

c. La précision par rapport aux nombre des ancrés

Le graphe ci-dessous montre l'erreur de localisation par rapport le nombre des ancrés :

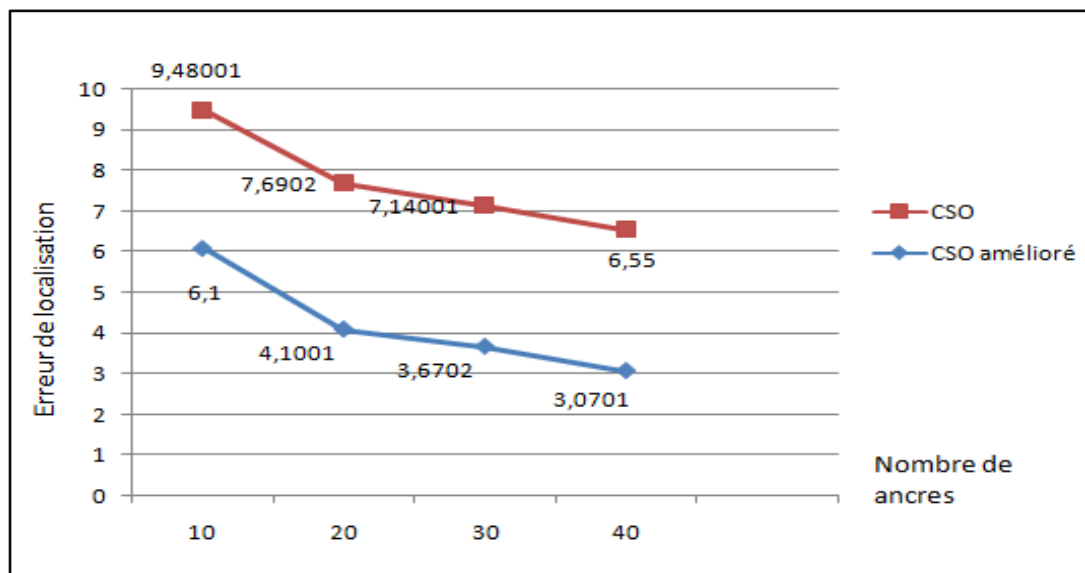


Figure IV.11 : Erreur de localisation Vs nombre des ancrés

Cette figure IV.11 montre clairement que la valeur de l'erreur moyenne diminue avec l'augmentation de nombre des ancrés pour les deux algorithmes : CSO et CSO amélioré.

d. La précision par rapport la taille de population :

Le graphe ci-dessous montre l'erreur de localisation par rapport la taille de population :

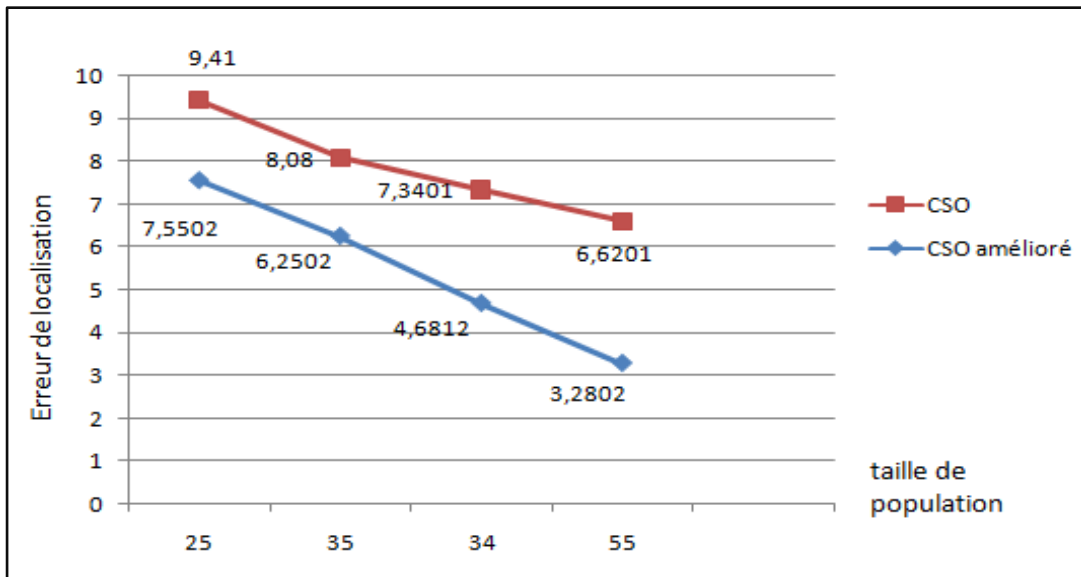


Figure IV.12 : Erreur de localisation Vs la taille de Population

D'après la figure IV.12, on voit que l'erreur moyenne diminue avec l'augmentation la taille de population pour les deux algorithmes : CSO et CSO amélioré. et on voit aussi que l'erreur moyenne de l'algorithme CSO amélioré est plus petit par rapport l'algorithme CSO.

e. La précision par rapport le nombre d'itération

Le graphe ci-dessous montre l'erreur de localisation par rapport le nombre d'itération :

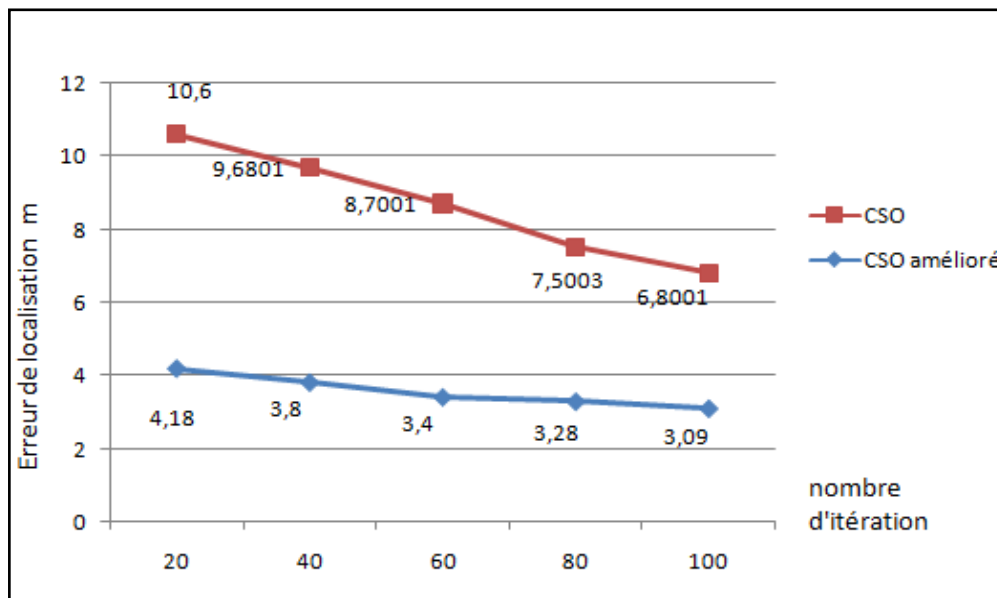


Figure IV.13 : Erreur de localisation Vs nombre d'itération

Cette figure IV.13 montre clairement que l'erreur moyenne diminue avec l'augmentation le nombre d'itération pour les deux algorithmes : CSO et CSO amélioré.

8. Comparaison :

Pour tester et vérifier les performances de notre proposition on comparer leur performances avec la méta-heuristique PSO présente dans [33]

Le tableau ci-dessus représente les paramètres fixes pour la comparaison :

Paramètres de réseaux		Paramètres de CSO			Paramètres de PSO	
paramètres	valeurs	paramètres	valeurs		paramètres	valeurs
zone de surveillance	100*100	Nombre d'itération	50		Nombre d'itération	50
Nombre des capteurs	100	La taille de population=40			W1	0.9
		coqs	poules	poussions	W2	0.2
					r1	2
Rayon de connectivité	40	9	13	18	r2	2

Tableau IV.2 : Les paramètres de comparaison

La figure IV.13 présente la comparaison entre les trois algorithmes CSO, CSO améliorée et l'algorithme PSO

Le tableau ci-dessus représente l'erreur de localisation par rapport le nombre des ancrs

Nombre des Ancres les algorithmes	10	15	20	25	30
	PSO	7	5.5	5.40	5.30
CSO	6.48	5.99	5.47	5.33	5.3
CSO améliore	2.36	2.3	2.36	2.2	2.05

Tableau IV .3 : Erreur de localisation Vs le nombre d'ancres

A travers la figure IV.13 qui indique l'erreur de localisation par CSO, CSO amélioré et PSO par rapport le nombre des ancrés, il est claire que la valeur de l'erreur de localisation est diminue avec l'augmentation de nombre de ancrés, dans les trois algorithmes

On remarque que les valeurs d'erreur de localisation de l'algorithme CSO est supérieur au algorithme PSO et CSO améliore lorsque le nombre d'ancres égale 15, puis devient presque égal à celle de PSO. Et on peut voir aussi que la moyen d'erreur de CSO améliore est inferieur par rapport les deux autres algorithmes

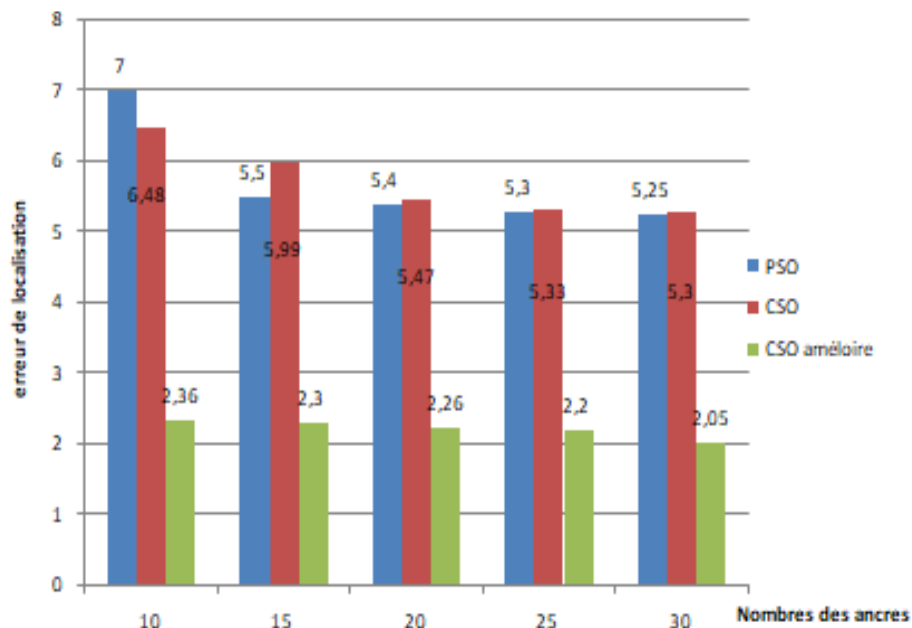


Figure IV.13 : la comparaison entre les trois algorithmes CSO, PSO et CSO amélioré

9. Conclusion :

Notre simulation montre que l'augmentation de la taille des essais améliore la précision de la localisation. En ajustant les paramètres de CSO avec les valeurs appropriées, nous pouvons obtenir un résultat optimal. La performance peut également être améliorée en augmentant l'itération maximale mais aussi il augmentera le temps de calcul.

De plus, le CSO amélioré est capable de fournir une solution optimale globale avec un nombre minimal d'itérations. Afin d'évaluer notre algorithme, Les résultats obtenus par CSO amélioré, dans la plupart des cas, fournissent des résultats meilleurs que PSO et la version standard de CSO.

Nous avons considéré le problème de localisation des nœuds dans un espace bidimensionnel. Cependant, dans un environnement réel, les nœuds de capteurs sont répartis dans un espace tridimensionnel. Mener des recherches sur l'algorithme proposé dans un espace tridimensionnel. De plus, l'algorithme proposé peut être étendu pour la localisation de nœuds dans un réseau de capteurs sans fil mobiles.

Conclusion générale

Les réseaux de capteurs constituent un axe de recherche très fertile et peuvent être appliqués dans plusieurs domaines différents. Cependant, il reste encore de nombreux problèmes à résoudre dans ce domaine afin de pouvoir les utiliser dans les conditions réelles. L'un des problèmes qu'on peut rencontrer dans ce genre de réseau est la localisation des capteurs sans fil. Plusieurs recherches ont été faites pour la conception des algorithmes qui tiennent compte de cette contrainte et en plus qui optimisons la précision de la localisation des nœuds dans ce type de réseau sans fils.

Dans ce travail, nous avons contribué à la résolution du problème de la localisation dans les réseaux de capteurs sans fil par l'adaptation d'une nouvelle méta-heuristique récemment inventé pour résoudre les problèmes d'optimisation combinatoire appelé l'algorithme d'essaim de poulets (CSO).

Nous avons implémenté et simulé l'algorithme CSO adapté à la localisation dans le langage de simulation MATLAB, et nous proposons une version amélioré de CSO pour couvrir le problème des nœuds non localisé à cause d'un nombre moins de trois ancrés dans leur voisinage, par considère certains nœuds dans laquelle leur emplacements ont été très précisément identifiés comme ancrés estimés.

Afin de montrer la performance de nos contributions une étude expérimentale est effectuée ainsi que Les résultats de simulations sont comparé avec ceux de l'algorithme PSO.

Les différentes évaluations effectuées montrent que l'algorithme CSO est très influencé par les paramètres de réseaux et de méthode, et nous a également permis de constater leur supériorité par rapport au PSO en terme de précision de la localisation. Pour les prochains travaux nous conseillons les chercheurs à proposer des versions parallèles de ce type d'algorithme afin de gagner le temps d'exécution et appliquer dans un environnement trois dimensionnelle et dans les réseaux de capteur mobile.

Référence :

- [1] I.F. Akyildiz, Et Al. "Wireless Sensor Networks : A Survey. Computer Networks (Elsevier)", Vol.38, No.4, March 2000, Pp.393- 422.
- [2] Mr Sahraoui Belkheyr, « Etude D'un Protocole De Routage Basé Sur Les Colonies De Fourmis Dans Les Réseaux De Capteurs Sans Fil », 2013.
- [3] Mlle.Oualad Ben Said Leyla & Mlle Bouregaa Fatima Zohra « Approche Hybride Pour L'optimisation De Consommation D'énergie Dans Les Réseaux De Capteur Sans Fil (Rcsf) » Mémoire Pour L'obtention Du Diplôme De Licence , 2015
- [4] Akyildiz, I.F.,W. Su, Y. Sankarasubramaniam, E. Cayirci,"A Survey onSensor Networks", IEEE Communications Magazine, August,102-114(2002).
- [5] Qinghuawanget Ilangkobalasingham,"Wireless Sensor Networks – An Introduction" Wireless Sensornetworks: Application-Centric Design, Geoff V Merrettand Yen Kheng Tan (Ed.), Isbn: 978-953-307-321-7, Intech,2010.
- [6] Ben L. Titzeret J. Palsberg, "Nonintrusiveprecision Instrumentation Of Microcontroller Software" , Acm, New York, Ny, Etats-Unis , Volume 40 Issue7,Pages 59 – 68, July 2005.
- [7] : H. Karl Et A. Willig, "Protocols And Architectures For Wireless Sensornetworks", John Wiley& Sons, 2005.
- [8] A. Montoya, D. C. Restrepoet D.A. Ovalle, "Artificial Intelligence For Wirelesssensor Networksenhancement",Intech,2010.
- [9] B. Krishnamachari, "Networking Wireless Sensors", Cambridge University press,2005.
- [10] David Martins, « Sécurité Dans Les Réseaux De Capteurs Sans Fil Stéganographie Et Réseaux De Confiance », L'u.F.R. Des Sciences Et Techniques De L'université De Franche-Comté, 2010.
- [11] Boubichedjallel Eddine. Une Approche Inter-Couches (Cross-Layer) Pour La Sécurité Dans Les R.C.S.F. Université De Batna, Faculté Des Sciences.
- [12] :Wassim Znaidi , « Modélisation Formelle De Réseaux De Capteurs À Partir De Tinyos »,Projet Fin D'étude, École Polytechnique De Tunisie,2006
- [13] Antoine Gallais, Jean Carle Et David Simplot-Ryl, « La K- Couverture De Surface Dans Les Réseaux De Capteurs », Ircica/Lifl, Univ.Lille 1, Inria Futurs

- [14] Boukhari Nawel, Aici Nacira, « Auto-Localisation Dans Les Réseaux De Capteurs Sans Fil », Memoire De Master, Université Mohamed Boudiaf - M'sil
- [15] Sari Mounya Amal « Etude Du Rssi Pour L'estimation De La Distance Dans Les Réseaux De Capteurs Sans Fil » Mémoire De Master 2017 Université Abou Bakrbelkaid–Tlemcen
- [16] Abdelkader Toudji Et Abdallah Dlimi « Une Approche Hybride Pour Conception D'un Protocole De Routage Multi-Saut Dans Les Réseaux De Capteur Sans Fil » Mémoire De Master Université Ahmed Draia – Adrar 2017
- [17] Sahraoui belkheyr « La Géo-Localisation Dans Les Réseaux Decapteurs Sans Fil. » Mémoire De Master 2011 université Abou Bakrbelkaid– Tlemcen
- [18] Boukhari Nawel « Auto-Localisation Dans Les Réseaux De Capteurs Sans Fil » Mémoire De Master Université Mohamed Boudiaf - M'sila
- [19] Mme Labraoui Nabila « La Sécurité Dans Les Réseaux Sans Fil » Mémoire De Doctorat ,Université Abou Bakrbelkaid–Tlemcen
- [20] Van Der Haegen, M. « Réseaux De Senseurs Sans Fil : Problèmes De Localisation ». Mémoire De Fin D'étude En Informatique. Université Libre De Bruxelles. 2007.
- [21] C. Gao, Z. Zhen, and H. Gong, “A self-organized search and attack algorithm for multiple unmanned aerial vehicles,” *Aerospace Science and Technology*, vol. 54, pp. 229–240, 2016.
- [22] Xianbing Meng, Yu Liu, Xiaozhi Gao, Hengzhen Zhang, “A new bio inspired algorithm: Chicken Swarm Optimization,” *Advance in swarm intelligence*, Vol: 8794, pp. 86-94, 2014.
- [23] CHALANE Idir. et OUARI Tarek. « Optimisation des paramètres d'un PID par essaims particulaires(PSO). » mémoire de Master Université A. MIRA de Bejaia 2017
- [24] BENDAHOUA Sarah « rapport sur le recuit simulé » Université d'Oran - Mohamed Boudiaf 2012/2013
- [25] Peng Chenet, Yongyi Mao « , Wireless sensor network node localization algorithm based on chicken swarm optimization and multi-power mobile anchor »
- [26] www.poules-club.com/la-hierarchie-chez-les-poules-histoire-de-comportement
- [27] www.google.com consulte 25/05/2019

[28] www.researchgate.net/publication/278691165_A_New_Bioinspired_Algorithm_Chicken_Swarm_Optimization

[29] Yongtao Li; Yu Wu; etXiangju Qu « Chicken Swarm–Based Method for Ascent Trajectory Optimization of Hypersonic Vehicles »

[30] www.slideshare.net/abdonajmeldin/chicken-swarm-optimization-cso

[31] Khaled Sayed Ahmed, Aboul Ella Hassanien, Siddhartha Bhattacharyya « A novel chaotic chicken swarm optimization algorithm for feature selection »

[32] Jingang Cao « A Localization Algorithm Based on Particle Swarm Optimization and Quasi-Newton Algorithm for Wireless Sensor Networks » *Department of Computer, North China Electric Power University, Baoding 071003, China*

[pso]

[34] N. Siddique and H. Adeli, "Nature Inspired Computing: An Overview and Some Future Directions," *Cognitive Computation*, vol. 7, no. 6, pp. 706–714, Nov. 2015