

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
UNIVERSITE FERHAT ABBAS - SETIF 1
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE.

MEMOIRE DE MASTER

DOMAINE : Mathématiques et Informatique

FILIERE : Informatique

SPECIALITE : Réseaux et Systèmes Distribués

Thème

Paradigmes bio-inspirés pour la
modélisation de la mobilité de sink
dans les réseaux de capteurs sans fil

Présenté par :

- BENZINE Ahmed Redha.

Dirigé par :

- Dr MECHTA Djamila.

Promotion : juin/2018

Remerciement

Mes premiers remerciements vont à Dieu pour m'avoir donné santé et volonté afin de réaliser ce travail.

Ensuite, Je tiens à remercier mon encadreur **Mme. MECHTA** enseignante au sein de l'Université FERHAT ABBAS, pour son encadrement, ses recommandations précieuses et sa disponibilité tout au long de ce projet.

Je tiens également à remercier les membres du jury d'avoir assisté à cette soutenance.

Enfin, je remercie tous les professeurs et enseignants pour leurs savoir et connaissances qu'ils m'ont apportées, ainsi que mes amis et camarades pour leurs encouragements et critiques envers ce travail.

Dédicace

Je dédie ce mémoire à :

- Ma mère et mon père, qui ont œuvrés pour ma réussite, de par leur amour, leur soutien, tous les sacrifices consentis et leur précieux conseils, pour toute leur assistance et leur présence dans ma vie, recevez à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.
- Mes frères qui n'ont cessés d'être pour moi des exemples de persévérance, de courage et de générosité.
- Mes amis les plus proches qui m'ont toujours soutenu, ainsi que mes camarades de classes avec qui j'ai traversé cette épreuve.

BENZINE Ahmed Redha.

Table des matières

Remerciement.....	I
Dédicace	II
Introduction générale.....	6

Chapitre 1 : Schémas bio-inspirés de la mobilité de la BS

1.1. Introduction	8
1.2. La mobilité dans les RCSFs	8
1.3. Avantages de la mobilité	9
1.4. Station de base mobile.....	9
1.5. La mobilité de la station de base	10
1.5.1. Schémas de mobilité classiques de la BS	10
1.5.2. Schémas de mobilité basés sur des méthodes bio-inspirés	11
1.5.2.1. Un algorithme de clustering basé sur l'algorithme bio-inspiré de colonie de fourmis basé pour les réseaux domestiques :.....	12
1.5.2.2. Algorithme optimale bio-inspiré des hormones (Bio Inspired Optimal Relocation of Mobile Sink Nodes in Wireless Sensor Networks)	14
1.5.2.3. Algorithme du Particle Swarm Optimisation (PSO)	16
1.5.2.4. Algorithme de colonie d'abeilles artificielles pour les RCSFs clairsemés	18
1.5.2.5. Algorithme des essaims de poissons artificiels AFSA (Prolonging WSN lifetime using a new scheme for Sink moving based on Artificial Fish Swarm Algorithm).....	20
1.5.2.6. Algorithme génétique (Genetic algorithm based length reduction of Mobile BS paths in WSNs)	23
1.6. Conclusion.....	26

Chapitre 2 : Schémas proposés pour la mobilité de la station de base

2.1. Introduction	27
2.2. Contributions	27
2.2.1. Firefly Based Energy (Firefly-BE)	27
2.2.1.1. Hypothèses	27
2.2.1.2. Principe.....	28
2.2.2. Wolf Search Sink Moving (WS-SM)	33
2.2.2.1. Principe.....	33
2.3. Conclusion.....	40

Chapitre 3 : Résultats et discussions

3.1. Introduction	41
3.2. L'environnement de développement	41
3.3. Simulation et évaluation	41
3.3.1. Les paramètres de simulation	41
3.3.2. Comparaison des performances.....	42
3.4. Conclusion.....	55
Conclusion générale	56
Bibliographie	57

Table des figures

Figure 1 : Organigramme de mobilité de BS basé sur ACO.	14
Figure 2: dessin explicatif des étapes des algorithmes proposés.	28
Figure 3: Organigramme du Firefly-BE.	30
Figure 4: L'algorithme Firefly-BE.	31
Figure 5: Représentation du comportement de chasse	35
Figure 6: Représentation du parcours cardinal.	36
Figure 7: Organigramme du WS-MS.	37
Figure 8: L'algorithme WS-MS.	38
Figure 9: Capture d'écran lors de l'initialisation du réseau.....	39
Figure 10: Capture d'écran lors de l'instant $t = 1000s$	39
Figure 11: Les divers cas de notre étude.	42
Figure 12: Nombre de nœuds survivants en gbest PSO et FFWS-MS ($A = 100 \times 100$ et $N = 100$).	43
Figure 13: Energie consommée en gbest PSO et FFWS-MS ($A = 100 \times 100$ et $N = 100$).	44
Figure 14: Nombre de données envoyées en gbest PSO et FFWS-MS ($A = 100 \times 100$ et $N = 100$).	44
Figure 15: Nombre de nœuds survivants en gbest PSO et FFWS-MS ($A = 100 \times 100$ et $N = 200$).	46
Figure 16: Energie consommée en gbest PSO et FFWS-MS ($A = 100 \times 100$ et $N = 200$).	47
Figure 17: Nombre de données envoyées en gbest PSO et FFWS-MS ($A = 100 \times 100$ et $N = 200$).	47
Figure 18: Nombre de nœuds survivants en gbest PSO et FFWS-MS ($A = 500 \times 500$ et $N = 100$).	49
Figure 19: Energie consommée en gbest PSO et FFWS-MS ($A = 500 \times 500$ et $N = 100$).	50
Figure 20: Nombre de données envoyées en gbest PSO et FFWS-MS ($A = 500 \times 500$ et $N = 100$).	50
Figure 21: Nombre de nœuds survivants en gbest PSO et FFWS-MS ($A = 500 \times 500$ et $N = 200$).	52
Figure 22: Energie consommée en gbest PSO et FFWS-MS ($A = 500 \times 500$ et $N = 200$).	53
Figure 23: Nombre de données envoyées en gbest PSO et FFWS-MS ($A = 500 \times 500$ et $N = 200$).	53

Liste des tableaux

Tableau 1: Tableau des différents paramètres de simulation.	41
Tableau 2 Table des paramètres	43

Introduction générale

Les réseaux de capteurs sans fil (RCSFs) sont des réseaux de communications autonomes idéals pour les tâches de détection d'évènements particuliers. Les capteurs sans fil qui constituent ce type de réseau, sont des appareils de taille miniature dotés d'une architecture adaptée à leur petite taille. Cette architecture se compose essentiellement d'une mémoire pour stocker des données, d'un CPU pour le traitement de ces données et d'une batterie, source de vie primordiale pour garder le capteur en fonction et permet d'effectuer les opérations de transmission : envoie et réception des données d'un capteur à un autre.

Le cycle de vie d'un capteur se résume souvent à capter des événements extérieurs, tels que des températures, une surveillance, des prévisions de catastrophes naturelles ... etc. Par la suite, ces données sont traitées et envoyées à une station de base (BS) fixe, qui à son tour, les transmet à un ordinateur central afin de les analyser.

Souvent exploités dans des zones hostiles, ou bien inaccessibles tels que les champs de batailles pour objectif d'espionnage, ou encore sur des sommets de volcans afin d'étudier les changements de températures, le renouvellement des batteries des capteurs devient une tâche difficile et parfois impossible, ce qui nécessite l'adoption de stratégies d'optimisation visant à réduire la consommation d'énergie des capteurs et ainsi augmenter la durée de vie du réseau.

Parmi les nombreuses stratégies proposées dont ceux qui visent à optimiser le routage de données ou le clustering, on distingue particulièrement le facteur de mobilité qui consiste en l'utilisation d'une BS mobile qui se déplace vers les nœuds capteurs pour but de réduire la distance entre entités. Afin de mouvoir la BS dans une zone, cette dernière doit suivre un schéma de mobilité qui trace le chemin à parcourir selon des critères précis.

Récemment, les techniques d'optimisation bio-inspirées ont attiré la curiosité des chercheurs par leurs rapidités de convergence vers la solution optimale, ce qui les a incités à adapter ces algorithmes en schémas de mobilité pour la BS. De nombreux schémas ont été développés et depuis, ont connus un grand succès garantissant une amélioration de la qualité de service du réseau visant à réduire considérablement l'énergie consommée.

Nous prenons exemple sur le succès des méthodes bio-inspirées afin de proposer à notre tour, un schéma de mobilité de la BS inspiré de deux algorithmes d'optimisation bio-inspirés (l'algorithme des lucioles et l'algorithme de recherche des loups).

Afin de mener à bien notre étude, notre projet s'organise selon le plan suivant :

- Le premier chapitre décrit l'efficacité de la mobilité de la BS dans les RCSFs ainsi que ses avantages. On présente ensuite, un état de l'art qui cite les modèles de mobilité bio-inspirés de la BS présents dans la littérature.
- Le second chapitre décrit nos contributions, un descriptif détaillé sur notre nouveau schéma de mobilité FFWS-MS qui permet d'optimiser la formation de cluster et la mobilité de la BS.
- Le troisième et dernier chapitre regroupe une série de simulations et de tests comparatifs du schéma *gbest* PSO et de notre schéma proposé pour prouver son efficacité dans la réduction de consommation d'énergie.

1.1. Introduction

Les réseaux de capteurs sans fil (RCSF) ont gagné en popularité et se voient utiliser beaucoup plus fréquemment. Le critère de prolongement de la durée de vie du réseau est primordial et dépend de la consommation d'énergie des capteurs. Afin d'augmenter la durée de vie des réseaux de capteurs, des chercheurs ont proposé de nouveaux mécanismes basés sur la mobilité des RCSFs.

Les réseaux de capteurs sans fil mobile sont une classe spéciale des RCSFs, dans laquelle un ou plusieurs composants du réseau sont mobiles. Ce composant mobile peut se déplacer dans le réseau et collecter des informations directement depuis les capteurs pour ensuite les transmettre à la station de base [1].

Il a été démontré que l'utilisation de la mobilité dans les RCSFs peut augmenter la durée de vie du réseau et améliorer leur performance grâce à l'équilibrage de charges entre les nœuds.

1.2. La mobilité dans les RCSFs

Dans certains cas où les nœuds sont mobiles, la mobilité peut être utilisée comme outil pour réduire la consommation d'énergie. Dans un réseau de capteurs statiques, les paquets provenant des nœuds suivent des chemins multi-sauts vers la station de base. Ainsi, certains chemins peuvent être chargés (sollicités plus que d'autres) et les nœuds proches de la Station de Base relayent plus de paquets [3]. Ils sont plus sujets à l'épuisement prématuré de leurs batteries. Si certains nœuds (éventuellement, la station de base) sont mobiles, le trafic peut être modifié si les nœuds mobiles sont chargés de collecter des données directement à partir de nœuds statiques.

Les stratégies reposant sur la mobilité peuvent être classées en deux ensembles : les stratégies avec un Sink mobile et les stratégies avec des relais mobiles, selon le type de l'entité mobile. Il est important de souligner ici que lorsque nous examinons des systèmes mobiles un problème important est le type de contrôle de la mobilité des nœuds qu'intègre la conception du réseau.

Les nœuds mobiles peuvent être divisés en deux catégories : ils peuvent être spécifiquement conçus comme partie de :

- L'infrastructure, leur mobilité peut être entièrement contrôlée dans la mesure où ils sont généralement robotisés.
- Lorsque les nœuds mobiles font partie de l'environnement, ils pourraient ne pas être contrôlables.

S'ils suivent un horaire strict, ils ont une mobilité complètement prévisible (par exemple, une navette pour les transports publics [4]). Sinon, ils peuvent avoir un comportement aléatoire de sorte qu'aucune hypothèse ne puisse être faite sur leur mobilité. Enfin, ils peuvent suivre un schéma de mobilité qui n'est ni prévisible ni totalement aléatoire.

1.3. Avantages de la mobilité

La mobilité dans les RCSFs est considérée comme un grand atout, nous citons dans ce qui suit les avantages de son utilisation [5] :

- **Amélioration de la durée de vie** : Les nœuds capteurs ont pour rôle de capter et de transmettre des données, la communication entre entités consomme énormément d'énergie. En réduisant la distance (multi-sauts) entre l'émetteur et le récepteur on augmente la durée de vie du réseau en préservant la batterie des nœuds.
- **Connectivité** : La mobilité permet aux nœuds de couvrir une large surface des zones les plus inaccessibles et des régions isolées.
- **Équilibrage de charges** : Les nœuds les plus proches de la BS représentent le goulot d'étranglement dans la transmission des données. Une BS mobile change constamment le nœud le plus proche et résout donc ce problème.
- **Exploration et localisation** : La mobilité permet aux nœuds de se mouvoir dans le réseau afin de réaliser l'exploration d'un site, réparation de la couverture et localisation d'une cible. Ceci est possible grâce à l'algorithme qui réunit les nœuds mobiles et statiques.

1.4. Station de base mobile

La station de base (aussi appelé puit) est un nœud différent des autres nœuds du réseau, elle joue le rôle de récolteur de données reçues de la part des nœuds déployés dans le RCSF. Il est connu que la station de base est statique dans une architecture de RCSF

classique. L'inconvénient majeur de cette architecture réside dans le goulot d'étranglement lorsque les nœuds proches envoient leurs données.

Afin de résoudre ce problème, et en sachant que la BS est une entité équipée d'une grande puissance de calcul, capacité de stockage et d'une énergie importante ; l'utilisation d'une BS mobile est la solution où contrairement à un RCSF classique, la BS peut se déplacer pour tenter de trouver un chemin optimal vers des nœuds spécifiques et ainsi diminuer la communication multi-sauts.

1.5. La mobilité de la station de base

1.5.1. Schémas de mobilité classiques de la BS

La première idée de l'utilisation d'une station de base mobile pour les RCSFs a été proposée par Shah et al. [6] où la BS est appelée « MULE ». La MULE parcourt le réseau aléatoirement afin de collecter les données émises par les capteurs. Ensuite, la MULE traite les données reçues pour les envoyer à un point d'accès filaire. Ce schéma était nouveau et facile à mettre en œuvre. La consommation d'énergie des nœuds de capteurs peut être significativement réduite car la plage de transmission entre le nœud et la MULE est significativement réduite.

Ma, Yang et Zhao [7] ont proposé un nouveau mécanisme de collecte de données pour les RCSF plus large en exploitant la mobilité de la BS. La station mobile, appelée M-collector ici, débute la collecte de données périodiquement en plusieurs tours. Elle traverse le réseau de nœuds en collectant les données depuis les capteurs proches pour ensuite les transmettre au sink une fois retourné au point de départ. Pour les applications avec distance stricte et/ou contrainte de temps, les auteurs ont également proposé un algorithme amélioré qui utilise plusieurs M-collectors. Chaque M-collector parcourt pendant un demi-tour au lieu d'un tour complet.

Le Set Packing Algorithm and TSP (SPAT) [8] a été proposé pour permettre une fiabilité et collecte des données efficace pour un capteur sans fil typique et réseaux d'actionneurs (WSANs). SPAT est basé sur le Set Packing Algorithm et TSP afin d'accéder à tous les capteurs dans le réseau cible de manière efficace. En utilisant SPAT, les utilisateurs sont en mesure de contrôler tous les hôtes de capteurs d'une manière efficace et en temps réel.

Konstantopoulos, Pantziou and Gavalas [9] ont proposés le protocole MobiCluster pour minimiser la consommation d'énergie. Dans ce protocole, les CHs collectent les données depuis les capteurs et effectuent après un filtrage de données sur les données brutes collectées. Ensuite, ils transmettent leurs données à des nœuds appelés Rendez-vous Nodes (RNs) qui se situent à proximité de la trajectoire de la BS mobile. La BS mobile parcourt ensuite le réseau et collecte les données depuis les RNs.

Gu, Ji et Li [10] ont proposé un travail afin d'analyser le problème de la mobilité de la BS avec contraintes de retard pour les RCSFs. D'autres problèmes tels que le zero delay, infinite speed et connected sink sites ont été discutés et des algorithmes optimaux furent développés. Les auteurs ont généralisé leurs solutions et ont proposé un algorithme optimal en temps polynomiale pour résoudre les problèmes des RCSFs.

Wang et al. [11] ont proposé un algorithme d'optimisation combinant le clustering et le déploiement de la BS pour les réseaux domestiques intelligents. Les auteurs ont proposé deux algorithmes sur la mobilité de la BS basés sur le clustering. Ces deux algorithmes sont EMCA et MECA, ils réduisent la moyenne de consommation d'énergie et prolongent la durée de vie du réseau. Plusieurs paramètres tels que le nombre de BS mobile, vitesse et trajectoire en mouvement ont été soigneusement étudiés avec des résultats de simulation et comparaison afin de démontrer une amélioration du protocole LEACH.

1.5.2. Schémas de mobilité basés sur des méthodes bio-inspirés

Les RCSFs souffrent de plusieurs problèmes majeurs tels que le déploiement aléatoire des nœuds, la sécurité, l'amélioration de la durée de vie, la consommation d'énergie, etc.

Les techniques basées sur les comportements naturels ont pas mal inspiré les chercheurs et ont pu résoudre la majorité de ces problèmes. Ces techniques sont basées sur le comportement des animaux, les bactéries, les plantes et même sur la génétique.

Comme nous l'avons cité précédemment, la station de base mobile peut être contrôlée ou non. Sur ce qui suit, nous allons citer quelques algorithmes basés sur des méthodes bio-inspirés pour les RCSFs qui améliorent la performance du réseau tout en utilisant une BS mobile.

1.5.2.1. Un algorithme de clustering basé sur l'algorithme bio-inspiré de colonie de fourmis basé pour les réseaux domestiques :

Jin Wang et al. [12] ont développé un algorithme qui combine un algorithme de clustering LEACH et l'algorithme de colonie de fourmis afin d'améliorer la durée de vie d'un réseau domestique. Cet algorithme est inspiré de la nature, on propose un algorithme de clustering basé sur un algorithme d'optimisation de colonie de fourmis (Ant Colony Optimization (ACO)) avec une station de base mobile ou le réseau est divisé en plusieurs cluster avec à chaque fois un cluster-head sélectionné. La station de base communique avec chaque cluster-head afin de collecter les données directement depuis une courte portée. L'ACO est utilisé pour trouver la trajectoire de mobilité optimale de la station de base.

1. Principe de fonctionnement

a. Le Clustering

Durant cette phase, le réseau est divisé en plusieurs clusters où chaque cluster possède un CH. Pour ce faire, on applique une variation de LEACH. Dans cet algorithme, les capteurs possédant la plus grande quantité d'énergie ont une grande probabilité d'être élus comme CHs. On détaille le processus dans ce qui suit :

Au début de chaque round, chaque nœud génère un nombre aléatoire entre 0 et 1, ensuite le compare à une valeur $T(n)$ prédéfinie. Si le nombre est inférieur à $T(n)$, et qu'il n'y a pas d'autres CHs dans la portée de transmission, alors ce nœud est élu un CH pour ce round. Toutefois, si un CH candidat est dans la portée de transmission, alors le candidat avec la plus grande quantité d'énergie devient CH et le deuxième devient un nœud membre. La valeur prédéfinie $T(n)$ est définie comme suit :

$$T(n) = \begin{cases} \frac{p}{1 - p^{*(r \bmod 1/p)}} \cdot \frac{E_{current}}{E_{initial}} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

Où P est le pourcentage désiré de CHs, r est le numéro de round, et G est le groupe de nœuds qui n'ont pas été sélectionnés comme CHs dans les derniers $1/P$ rounds. $E_{current}$ est l'énergie restante des nœuds et $E_{initial}$ est l'énergie initiale des nœuds.

Après s'être choisis comme CH, le nœud envoie par broadcast un message aux autres nœuds ordinaire. Ces nœuds vont rejoindre le CH selon l'intensité du signal et les clusters sont finalement formés. Les communications au sein d'un cluster peuvent être faites avec la méthode TDMA. Pour cela, chaque CH établie un Schedule TDMA pour ses membres, en indiquant pour chaque nœud son slot d'émission. Ce Schedule est envoyé aux membres par broadcast.

b. La mobilité de la BS basée sur ACO

Dans le but d'augmenter la durée de vie d'un réseau domestique, une station de base mobile est utilisée afin de traverser le réseau et collecter les données directement depuis les CHs. On remarque que la latence du réseau est ainsi diminuée. L'algorithme ACO est utilisé pour déterminer la stratégie de mobilité. Dans cet algorithme, la BS est traitée comme un vendeur et les CHs comme les destinations qu'il doit visiter. La BS mobile va emprunter le chemin optimal déterminé par l'ACO. Elle va visiter chaque CH afin de collecter les données en utilisant des communications physiques d'un seul saut. Sachant que la portée de transmission est relativement courte entre la BS et le CH, la consommation d'énergie des CHs diminue largement.

En résumé, les CHs sont d'abord choisis grâce au LEACH amélioré, ces derniers collectent les données depuis leurs nœuds membres. Ensuite, on applique l'ACO pour trouver le chemin optimal pour la BS mobile qui visite les CHs. L'organigramme de l'algorithme se présente comme suit :

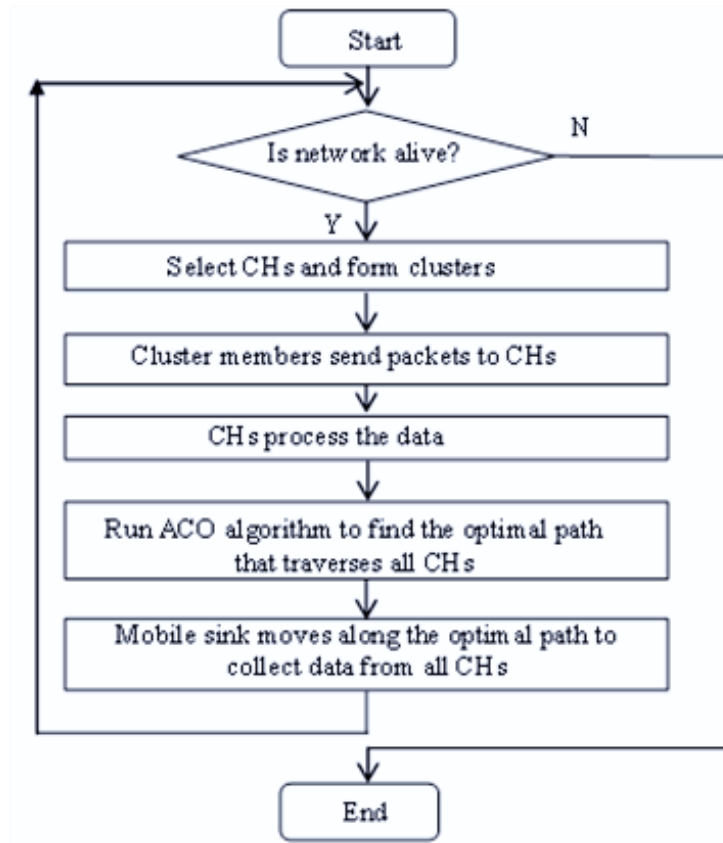


Figure 1 : Organigramme de mobilité de BS basé sur ACO.

2. Avantages

Après plusieurs simulations, les résultats prouvent que cet algorithme améliore les performances des réseaux domestiques lors de la mobilité de la station de base en termes de consommation d'énergie et augmente la durée de vie du réseau en comparaison avec d'autres algorithmes de routage déployés pour l'automatisation des réseaux domestiques tels que le standard LEACH et Mobile-P.

1.5.2.2. Algorithme optimale bio-inspiré des hormones (Bio Inspired Optimal Relocation of Mobile Sink Nodes in Wireless Sensor Networks)

Dans [13], Sumit Kataria et Aarti Jain proposent un algorithme qui optimise le déplacement des BS mobiles afin de préserver un certain équilibre d'énergie entre les nœuds. Ce déplacement est possible en utilisant le schéma bio-inspiré des Hormones. Dans cette méthode, la BS est guidée afin de se déplacer intelligemment vers l'emplacement optimal, ce qui améliore la durée de vie du réseau et contre balance la consommation d'énergie.

Après plusieurs simulations, on remarque que la durée de vie du réseau est améliorée en comparaison avec d'autres méthodes.

1. Principe de fonctionnement

L'utilisation des RCSFs nécessite un grand nombre de nœuds dans certains cas d'applications qui sont propagés dans des régions de plusieurs milliers de kilomètres. Dans ce cas, le réseau doit être scalable. Pour ce faire, le réseau est divisé en plusieurs clusters avec un CH pour chacun d'eux en appliquant un algorithme de clustering. Chaque CH est chargé de collecter les données des nœuds membres de son cluster afin de les transmettre à la BS. Cette technique fonctionne parfaitement pour les réseaux assez petits. Toutefois, dans le cas d'un grand réseau un nœud ordinaire n'est pas capable d'assurer toutes ces tâches. Dans notre cas, le réseau est divisé en clusters mais au lieu d'utiliser les techniques de clustering citées dans la littérature, une BS est assignée à chaque cluster comme étant un CH.

Le DHM (**Digital Hormone Model**) est inspiré du système biologique, où les cellules réagissent aux différentes hormones. La particularité des hormones est que les cellules peuvent communiquer sans pour autant connaître des identifiants ou adresses. L'idée est que les BS du réseau peuvent changer leurs liens dynamiquement pour se connecter avec les autres nœuds en se basant sur le DHM. Elles utilisent des messages « hormones » afin de communiquer, collaborer, et accomplir des comportements globaux. Les messages « hormones » ne portent pas d'adresses mais se propagent dans le réseau. Tous les nœuds suivent le même protocole. Lorsque les nœuds reçoivent les hormones, ils réagissent selon l'hormone reçu, leurs topologies locales et leurs informations. Les stations de bases communiquent avec les capteurs par le biais d'hormones et peuvent ainsi prendre des décisions efficaces pour de futures actions.

Pour résumer, l'idée de l'utilisation du DHM consiste à ce que les BS génèrent et diffusent des messages « hormones » dans leurs portés de transmissions. Les capteurs qui reçoivent ces hormones vont transmettre une réponse « hormones » à la BS en envoyant leurs coordonnées, leurs données et leurs énergie restante. Selon la réponse « hormones » reçu, les BS performant des actions afin de trouver le chemin optimal et se déplacent vers les positions des nœuds de chaque cluster pour les servir. L'algorithme se présente comme suit :

- **Etape 1 :** Nous avons N clusters où il existe un nœud sink mobile.
- **Etape 2 :** Le sink transmet un message hormone à tous les nœuds de son cluster, $S = \text{Steady State}$.
- **Etape 3 :** Le nœud sink va recevoir des réponses hormones depuis tous les nœuds connectés, $S = \text{Steady State}$.
- **Etape 5 :** Le sink va calculer g_k , g_k : (Gradient Move Rule) le sink dans l'état steady state collecte tous les hormones H_k qu'il reçoit depuis les capteurs, ensuite il calcule g_k avec : $g_k = r_i(t_x) / d_i * 1 / E_s * \text{LOAD}$.
- **Etape 5 :** $S = \text{move}$, le sink se déplace vers les capteurs dont la valeur de g_k est la plus grande (g_k^{\max}).
- **Etape 6 :** $S = \text{Steady State}$ lorsque $(x_k, y_k) = (x_i, y_i)$ sinon $S = \text{Move State}$, (x_k, y_k) : Les coordonnées de sink.
- **Etape 7 :** Retourner vers **Etape 1**.

2. Avantages

De nombreuses simulations ont été faites, et il a été remarqué que la durée de vie du réseau est grandement améliorée, mais aussi une consommation d'énergie plus efficace. Cependant, cet algorithme n'améliore pas les autres faiblesses des RCSFs telles que la latence, la consommation de bande passante et le rapport de livraison de données.

1.5.2.3. Algorithme du Particle Swarm Optimisation (PSO)

Les auteurs dans [14], ont introduit un protocole de mobilité pour la BS afin de prolonger la durée de vie du réseau et l'optimisation de l'énergie. Le protocole proposé est basé sur l'algorithme PSO. Cet algorithme s'inspire à l'origine du monde du vivant. Il s'appuie notamment sur un modèle développé par Craig Reynolds à la fin des années 1980, permettant de simuler le déplacement d'un groupe d'oiseaux. Cette méthode d'optimisation se base sur la collaboration des individus entre eux. Elle a d'ailleurs des similarités avec les algorithmes de colonies de fourmis, qui s'appuient eux aussi sur le concept d'auto-organisation. Cette idée veut qu'un groupe d'individus peu intelligents puisse posséder une organisation globale complexe.

Chapitre 01 : Schémas bio-inspirés de la mobilité de BS

Ainsi, grâce à des règles de déplacement très simples, les particules peuvent converger progressivement vers un minimum local. Le but principal de ce protocole est de chercher les coordonnées optimaux des sites que la BS doit visiter afin de collecter les données.

1. Principe de fonctionnement

On organise les nœuds en K clusters. La fonction fitness utilisée est basée sur la distance entre les capteurs et l'emplacement de la station de base.

La distance entre un nœud j et la station de base k est calculée comme suit :

$$d(m_{k,j}) = [(x_k - x_j)^2 + (y_k - y_j)^2]^{1/2}$$

Où : (x_k, y_k) sont les coordonnées de la station de base, et (x_j, y_j) sont les coordonnées du nœud j.

On doit minimiser la distance totale entre tous les nœuds et la station de base, la station de base doit se déplacer vers le site où cette somme est la plus petite.

$$f = \operatorname{argmin} \sum_{k=1}^K \sum_{j=1}^{|C_k|} d(m_k, n_j)$$

Où : $|C_k|$: est le nombre de nœuds qui appartiennent au cluster C_k .

f : est la fonction de fitness à minimiser avec l'algorithme PSO.

L'algorithme se déroule en deux phases :

The setup phase : Les capteurs sont dispersés aléatoirement, on suppose que la BS n'a pas d'informations sur l'emplacement des nœuds. Ainsi, tous les nœuds doivent envoyer les informations concernant leur emplacement à la station de base durant le début de cette phase.

En se basant sur ces informations, la BS applique l'algorithme PSO afin de déterminer les positions de points de collecte de données.

The steady phase : Lorsque la BS atteint sa destination, elle s'arrête pendant une certaine période t pour la collecte de données. Après la fin du Schedule TDMA dans un cluster, la BS se déplace vers un autre cluster pour collecter plus de données.

2. Avantages

Le protocole proposé peut prolonger considérablement la durée de vie du réseau en comparaison avec d'autres protocoles, aussi il permet d'augmenter la livraison de donnée chez la station de base.

1.5.2.4. Algorithme de colonie d'abeilles artificielles pour les RCSFs clairsemés

Dans les RCSFs clairsemés, un robot mobile (station de base) est généralement utilisé pour la collecte de données. Les capteurs possèdent une portée de transmissions et le robot mobile doit s'en approcher afin de recevoir les données de chaque capteur. Afin de minimiser la consommation d'énergie lors du mouvement du robot mobile, il est important d'établir un chemin de collecte de données minimal. Toutefois, il est essentiel que ce chemin atteigne toutes les portées des nœuds. De ce fait, il faut trouver la séquence des nœuds à visiter mais aussi le point de rendez-vous sur chaque portée.

Le robot doit être dans la portée de transmission d'un capteur avant de pouvoir entamer la collecte de données. La couverture d'un capteur peut être spécifiée par un disque dont le rayon est égal à la portée de transmission du capteur.

Zeng et al. [15] ont remarqué que ce problème peut être formulé comme étant un problème d'un voyageur avec des voisins (Traveling Salesman Problem with Neighborhoods), qui est un problème NP-Hard. Afin de résoudre ce problème, les auteurs ont proposé un algorithme DCCP (Data Collection Path Planning) basé sur la colonie d'abeilles (ABC)

1. Principe de fonctionnement

On suppose un point de départ S , on doit trouver un chemin valide pour que le robot mobile puisse visiter chaque disque au moins une fois afin de collecter les données depuis les capteurs pour ensuite revenir vers S . On définit P_1 qui est le premier point de rendez-vous où le robot mobile entre dans un disque C_i . Un chemin valide est représenté par la séquence de points de rendez-vous suivante : $R_i = \{S, P_1, P_2 \dots P_n, S\}$

Afin de minimiser la consommation d'énergie de routage du robot mobile, on doit chercher un chemin valide avec une distance totale minimale. Aussi, le problème DCCP est exprimé par la formule suivante : $\min_{R_i} = d(S, P_1) + d(S, P_n) + \sum_{j=1}^{n-1} d(P_j, P_{j+1}) \dots (1)$

Les étapes de l'algorithme ABC sont :

a. La phase d'initialisation

Tout d'abord, nous devons créer une population de la solution initiale, le chemin de routage initiale, selon la taille de la colonie S . Dans cette phase il y a deux étapes différentes :

- L'initialisation de la séquence visitée Seq des capteurs en ajoutant par itération le nœud le plus proche à Seq jusqu'à ce que l'on ajoute tous les nœuds. La distance est dé terminée par le point central de chaque disque d'un nœud.
- On génère des points de rendez-vous n depuis n extrémités de cercle comme points de rendez-vous initiaux. Lorsque l'on obtient un chemin initial R_j , on calcule sa valeur de fitness ϕ_i comme suit :

$$\phi_i = \frac{1}{d(S,P_1) + d(S,P_n) + \sum_{j=1}^{n-1} d(P_j,P_{j+1})} \dots \dots (2)$$

Après la phase d'initialisation, tous les chemins initiaux S sont créés. Le meilleur avec la plus grande valeur de fitness est enregistré comme ϕ_{best} .

b. La phase de recrutement des abeilles

Dans cette phase, S abeilles sont recrutées afin d'améliorer les S chemins initiaux R_i , en commençant par le second point dans R_i et en changeant la séquence de points de rendez-vous pour ensuite mettre à jour leurs positions. Après avoir mis à jour toutes les positions de points de rendez-vous, on obtient un nouveau chemin de routage R'_i . Ensuite, on calcule la valeur de fitness ϕ'_i de ce chemin avec (2), et on compare ϕ'_i avec l'ancienne valeur ϕ_i pour vérifier si R'_i est la meilleure solution, si c'est le cas, on enregistre ce chemin comme R_i afin de représenter le meilleur chemin trouvé.

2. Avantages

Cet algorithme permet effectivement une amélioration de la consommation d'énergie du robot mobile vu qu'il emprunte le chemin le plus court.

1.5.2.5. Algorithme des essaims de poissons artificiels AFSA (Prolonging WSN lifetime using a new scheme for Sink moving based on Artificial Fish Swarm Algorithm)

Les auteurs [16] ont, après plusieurs recherches sur les schémas bio-inspirés déjà traités, décidé de traiter le problème en se basant sur l'algorithme des essaims de poissons artificiels (AFSA) afin de tester son efficacité sur la mobilité de la BS dans les RSFCS.

Leur étude débute par l'analyse de l'algorithme de clustering LEACH-C. Les auteurs décident de proposer un protocole afin d'améliorer le LEACH-C. Ils ont défini les étapes suivantes pour chaque round :

- Division du réseau : consiste à diviser le réseau en différentes zones (de même taille).
- Formation géographique de Grids (grilles) et processus d'élection de Grid-Heads (GHs) : une élection de GH pour chaque zone.
- Mobilité de la BS : la BS se déplace vers la meilleure position en utilisant la méthode bio-inspiré AFSA.
- Transmission de données inter-grid : les GHs traitent les données reçus de la part de tous les nœuds de la grille. Les GHs envoient ces données à la BS en un seul saut.

La division du réseau en grilles : En début de chaque round, la BS se déplace vers le centre de la zone d'intérêt, et chaque nœud envoie un paquet de données contenant son identifiant, sa réserve d'énergie (E) et son emplacement (X, Y) dans le réseau. On partitionne le réseau en plusieurs grilles de tailles identiques, le but de ce partitionnement est l'équilibrage de charge efficace qui améliore considérablement la durée de vie du réseau mais aussi la performance de livraison de données.

La phase d'élection des GHs : l'élection de GHs se fait selon les critères suivants :

1. Les grid-heads sont sélectionnés sur la base de l'énergie résiduelle : Les nœuds avec une quantité d'énergie plus grande que la moyenne d'énergie de tous les nœuds d'une zone sont considérés comme étant candidats (éligible) de devenir GHs.

$$Current_Energy_i > AVG \dots (1)$$

AVG : Moyenne d'énergie du réseau, $Current_Energy_i$: Energy actuelle du ième nœud.

$$\sum_{i=1}^N Current_Energy_i / N \dots (2)$$

Où N : nombre des nœuds survivants.

2. Un GH est sélectionné sur la base de sa position dans la grille. Le nœud le plus proche du centre de densité de sa grille est le meilleur GH.

$$X_c = \frac{\sum_{i=1}^n X_i}{n}, Y_c = \frac{\sum_{i=1}^n Y_i}{n} \dots (3)$$

X_c : La coordonnée X du centre de densité de la grille $grid_i$

Y_c : La coordonnée Y du centre de densité de la grille $grid_i$

n : Le nombre de nœuds membres de la grille $grid_i$

Une fois que les GHs optimaux sont déterminés, la BS envoie un message par broadcast qui contient l'identifiant de chaque GH de chaque zone. Chaque nœud compare son ID avec l'ID du GH, s'il y a correspondance, alors il agit comme étant un GH. Sinon, le nœud détermine son slot dans le Schedule de transmission et envoie ses données au GH de son slot.

1. Principe de fonctionnement :

Etape 1 : Détermination et initialisation des paramètres.

On définit une population de poissons artificiels comme étant des nœuds capteurs dont le nombre total est N . On considère dans cet algorithme qu'il y a un **poisson artificiel mobile (MAF)** qui est la BS mobile et les autres AFs sont les nœuds capteurs de la zone. X est l'état courant de la BS (état initial dans le centre de la zone), $visual$ est le champ de vision de la BS avec une valeur assez large pour une recherche globale, donc le champ de vision dans notre algorithme est égal à la valeur de la diagonale de la grille, $pas = 1$ (pour une meilleure convergence), le nombre d'itérations est égale à 100, et le facteur de foule est variable selon la densité de la zone.

Etape 2 : Application de l'AFSA.

On s'inspire de l'idée basique du AF-Swarm et du AF-Prey afin de trouver la meilleure position (quasi-optimale) dans le champ de vision de la BS.

1. D'abord, on applique le AF-Swarm, en calculant la fonction de fitness F , qui représente dans notre algorithme la densité des nœuds dans chaque grille (nf). Les GHs qui sont situés dans le champs de vision de la BS ($d_{bs-GH} < visual$), et avec une valeur de ($nf \geq \delta$), où $\delta = 10$, on donne au facteur de foule la valeur de 10 qui est la valeur minimale de la densité d'une grille quand N (le nombre de nœuds de la zone) est supérieur ou égale à $(1/2 nn)$ (la moitié). Ce qui veut dire qu'un GH avec plus de nœuds membres (nf) et plus de donnée à agréger (concentration de nourriture dans AFSA), est considéré comme un GH candidat (éligible).

Chapitre 01 : Schémas bio-inspirés de la mobilité de BS

2. Si N est inférieur à la moitié, alors le facteur de foule δ sera inférieur, on lui attribue donc la valeur de 5 (valeur minimale de la densité), sinon s'il n'y a pas de GH qui satisfait les conditions de l'essaim, on exécute l'AF-Prey.
3. Après avoir employé l'AF-Swarm on obtient comme résultats une liste de GHs éligible, on calcule le centre $X_c(x_c, y_c)$ de la densité de ces GHs ou $X_c \in$ champs visuel de la BS ($d_{ic} < visual$), et on cherche la meilleure position (près des nœuds GHs avec le plus de capteurs) ou la BS va se déplacer (en direction de ce centre X_c) en utilisant la formule suivante (AF-Swarm):

$$X_v = X_i + \frac{X_c - X_i}{\|X_c - X_i\|} \times Step \times rand(0,1) \quad \dots(4)$$

$X_i(x_i, y_i)$: L'emplacement actuel de la BS.

$X_v(x_v, y_v)$: L'emplacement ou la BS se déplacera (pour se rapprocher du centre X_c).

$X_c - X_i$: un vecteur de transfert de X_c vers X_i , $X_c - X_i = (x_c - x_i) + (y_c - y_i)$

$\|X_c - X_i\|$: est la distance euclidienne entre la position actuelle X_i de la BS et la position du centre X_c .

$$\|X_c - X_i\| = (x_c - x_i)^2 + (y_c - y_i)^2$$

rand : fonction aléatoire qui génère un nombre aléatoire entre [0, 1].

4. Si la condition de l'AF-Swarm n'est pas satisfaite (il n'y a pas de GH qui satisfait la condition du facteur de foule ($nf < \delta$)), on exécute l'AF-Prey en utilisant la formule :

$$X_v = X_i + visual \times rand(-1,1)$$

$X_v(x_v, y_v)$: Le prochain emplacement dans le champs de vision de la BS ou cette dernière va se déplacer après comparaison de sa fonction de fitness avec celle de sa position actuelle X_i . La fonction de fitness est la distance totale entre X_v et chaque GH dans le champ de vision. Si la distance totale dans cette position est inférieure a la distance totale quand la BS est à l'emplacement X_i , alors c'est la meilleure position a laquelle la BS va se

déplacer en utilisant la formule (4). Sinon la BS choisie un autre X_v jusqu'à fin du nombre d'itérations.

Dans cette stratégie, la position optimale de la BS doit être proche des GHs qui couvrent une grande surface (un grand nombre de nœuds).

2. Avantages

Après simulation, l'algorithme AFSA-MS permet de trouver une solution quasi-optimale pour le positionnement de la BS qui devrait être le plus proche des GHs qui couvrent une grande zone, afin qu'ils couvrent le plus de capteurs. Egalement, les résultats de la simulation montrent que AFSA-MS consomme moins d'énergie ce qui maximise la durée de vie du réseau en maintenant le plus de nœuds en vie pour de longues durées, ce qui engendre une transmission de données maximale à la BS.

Le seul point négatif dans ce protocole est que le premier nœud meurt dans un court moment en comparaison avec LEACH-C.

1.5.2.6. Algorithme génétique (Genetic algorithm based length reduction of Mobile BS paths in WSNs)

Les algorithmes génétiques (Genetic Algorithms) sont des stratégies d'adaptation et des techniques d'optimisation globale [17]. Ce sont les premiers, les plus connus, et les plus utilisés parmi les méthodes évolutionnaires.

Les auteurs dans [18] ont tentés de résoudre les problèmes des RCSFs en utilisant la mobilité de la BS en raccourcissant le chemin parcouru au maximum. Ils ont minimisé les points d'arrêt de la BS, afin de réduire la consommation d'énergie des nœuds et alléger le problème de latence. Ils y sont parvenus grâce aux algorithmes génétiques.

1. Principe de fonctionnement

On suppose un RCSFs homogène, où tous les nœuds à l'exception de la BS sont immobiles.

- R : la portée de transmission entre la BS mobile et un nœud.
- r : la portée de transmission entre les nœuds.
- C : l'ensemble des points du nombre de n .
- D : l'ensemble de nœuds du nombre de m .

Chapitre 01 : Schémas bio-inspirés de la mobilité de BS

- P : les points d'arrêt résultants.

Etape 1 : Division du réseau en clusters

En utilisant un simple algorithme de clustering, où chaque nœud sélectionne son CH selon leur distance euclidienne.

Etape 2 : Réduction du chemin par GA :

- 1- On a un ensemble de points $D = \{d_1, d_2, \dots, d_m\}$, $C = \{c_1, c_2, \dots, c_m\}$
- 2- On doit minimiser $|P|$, ce qui implique que P doit être minimal.

Etape 3 : Construction du réseau virtuel

D'abord, on construit un réseau virtuel dans lequel on choisit l'emplacement où la BS doit aller. On choisit le point le plus à gauche, le plus à droite, le plus en haut et le plus en bas dans D. Ensuite, on dessine un rectangle depuis les quatre points choisis précédemment de telle sorte que ce rectangle contienne tous les points de D. On divise le rectangle en plusieurs petites grilles avec une longueur d'arête L ou $[0 < L \leq \sqrt{2}R]$.

Pour chaque point d_i de D, on calcule tous les points d'intersection du réseau de grilles que d_i peut atteindre avec une distance R. Les points d'intersection avec un grand nombre de voisins vont grandement contribuer à la couverture et à la connectivité du réseau, ils ont donc une plus grande priorité pour être choisis, en d'autres termes avec une large *Neighbor – Count*.

Etape 4 : Sélection des points d'arrêt :

Par la suite, un modèle de poids a été découvert dans lequel chaque CH possède une valeur de poids basée sur le nombre de ses fils et sa localisation.

Neighbor-count_i : le nombre de points d'intersection des CHs voisins.

NeighborSet_i : (*ns_j*) Les CHs voisins d'un point d'intersections (l'ensemble des CHs qui atteint le point d'intersection avec une distance R) on a :

$$NeighborSet_i = \{ns_1, ns_2, \dots, ns_{Neighbor-count_i}\}$$

L'ensemble des nœuds fils d'un CH j :

$$ChildSet_j = \{child_1^j, child_2^j, \dots\}$$

Un CH avec un $ChildSet_j$ important, contribue grandement à la couverture du réseau et sa connectivité, donc le poids résultant doit être petit, et les petites distances entre les points d'intersections et leurs CHs voisins seront considérés en premier. Donc plus la distance est petite plus le poids est petit.

On définit le schéma de poids comme suit :

$$W_i = \sum_{ns_j \in NeighborSet_i} \frac{m. (Distance(c_i, ns_j) + 1)(r + \sum_{k=1}^{|childSet_j|} Distance(ns_j, child_k^j))}{r. (NeighborCount_i + 1)(|childSet_j| + 1)}$$

$$X_i = \{1, c_i \in P ; 0, c_i \notin P \quad i = \{1, 2, \dots, n\}$$

Avec la fonction objective (Fitness) est : $\min f = \min \sum_{i=1}^n W_i X_i$

Etape 5 : Construction de l'ensemble de points d'arrêt minimale

En se basant sur l'ensemble de points d'intersections obtenu précédemment et en utilisant l'algorithme GA qui utilise le croisement, on obtient l'ensemble de points d'arrêt minimaux.

Etape 6 : Construction du plus court chemin

Dans cette étape, le but principal est de construire le plus court chemin qui passe par tous les points d'arrêt en utilisant le deuxième GA avec *genetic degree combination crossover (ER)* et *displacement mutation (DM)*.

2. Avantages

Cet algorithme est l'un des plus efficaces du fait qu'il permet de se rapprocher de la solution optimale pour le déplacement de la BS, et cela même si le problème est grand.

1.6. Conclusion

Dans ce chapitre, nous avons expliqué les différents avantages que la mobilité peut apporter aux RCSFs, aussi nous avons cité l'emploi de cette mobilité dans un réseau de capteurs et plus précisément la mobilité de la station de base.

Après avoir vu les différents types de mobilité de la BS et après concertation de la littérature, nous avons expliqué plusieurs algorithmes bio-inspirés qui traitent la mobilité de la BS. Plusieurs articles scientifiques comparent leurs travaux bio-inspirés avec d'autres algorithmes classiques. Les résultats obtenus de ces comparaisons nous motivent à développer une approche bio-inspirée pour augmenter la durée de vie d'un réseau.

2.1. Introduction

Dans ce chapitre, nous allons présenter notre contribution le Firefly Wolf Search Moving Sink (FFWS-MS) qui consistent en un algorithme bio-inspiré pour prolonger la durée de vie d'un réseau de capteurs. Notre algorithme se divise en deux grandes étapes dont la première consiste au clustering inspiré de l'algorithme des lucioles (Firefly Algorithm), tandis que la seconde traite la communication tout en utilisant une BS mobile qui suit une trajectoire déterminée par l'algorithme de recherche des loups (Wolf Search Algorithm).

2.2. Contributions

2.2.1. Firefly Based Energy (Firefly-BE)

L'algorithme Firefly (FA) est une récente méthode d'optimisation globale stochastique méta-heuristique, aussi un algorithme basé sur l'intelligence de l'essaim. Yang [19] a développé l'algorithme inspiré du comportement social des lucioles et du phénomène de la communication bioluminescente. L'algorithme fonctionne essentiellement sur la base de deux facteurs : la distance entre les lucioles et l'intensité de lumière [20]. La sortie de l'algorithme est un ensemble de positions où est regroupé la population des lucioles les plus brillantes, selon la modélisation du problème.

2.2.1.1. Hypothèses

- Le réseau contient N nœuds fixes et homogènes, déployés aléatoirement dans un espace donné ($\text{AreaSize} \times \text{AreaSize}$).
- Seule la BS est Mobile et est placée au centre de la région, on considère également qu'elle possède une quantité d'énergie illimitée.
- Il existe deux types de nœuds : Cluster-Heads (CHs) et les nœuds ordinaires.
- Chaque nœud a son propre identificateur unique.
- Le nœud appartient à un seul Cluster-Head durant chaque round.
- Tous les nœuds ont initialement la même quantité d'énergie.
- Les CHs sont choisis aléatoirement dans chaque itération.
- Notre algorithme est basé sur un algorithme bio-inspiré pour faire le clustering : Firefly algorithm.

- L'algorithme se déploie sur un environnement centralisé dont toutes les procédures de clusterisation sont effectuées à la station de base.
- La simulation se poursuit jusqu'à ce que toute l'énergie soit consommée (ou bien qu'il n'y est plus assez de nœuds pour former des clusters) ou que la durée de simulation soit terminée.
- Pour adapter notre algorithme au contexte, nous avons considéré l'intensité de lumière comme étant l'énergie des capteurs.
- Les CH sont 5% de la population.

2.2.1.2. Principe

Cet algorithme se base sur les points suivants :

- 1- Le choix d'appartenance à un cluster est basé sur la distance entre le nœud et le cluster-Head.
- 2- L'énergie des nœuds est déterminée en fonction d'une fonction objective à optimiser (appelée fonction de coût).
- 3- Après un nombre d'itérations maximal déterminé, on obtient la particule des CHs optimale pour la communication.
- 4- La communication se fait en deux étapes : TDMA (Accès multiple à répartition dans le temps) et CDMA (Code division multiple access).

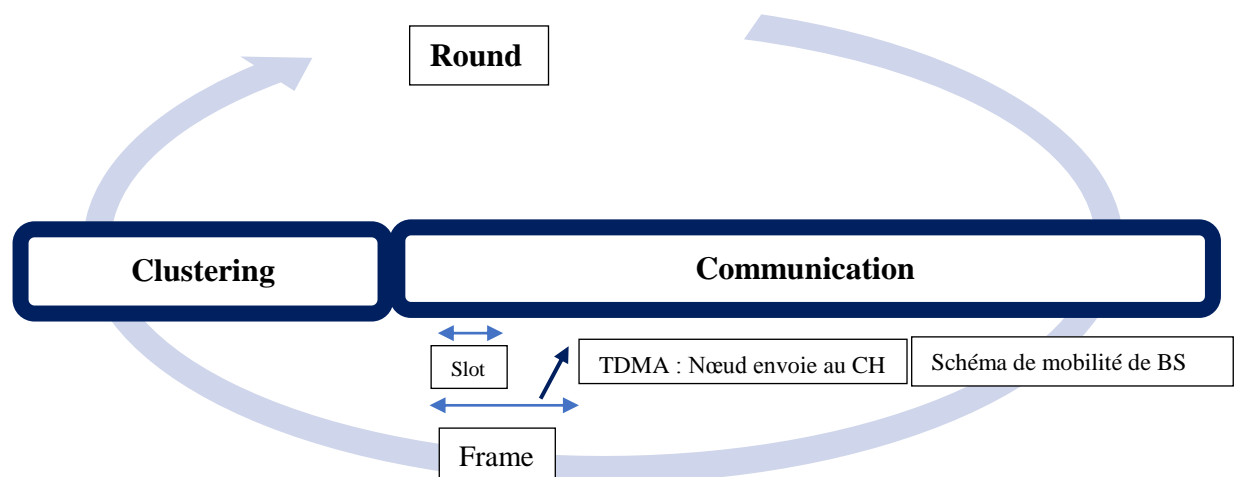


Figure 2: dessin explicatif des étapes des algorithmes proposés.

1- Phase 1 : Clustering

- Choisir les particules où chaque particule contient un nombre déterminé de CHs aléatoirement.
- Chaque nœud calcule la distance entre lui et toutes les particules (les CHs), et choisit le CH le plus proche pour lui rejoindre.
 - Pour la formule de calcul de la distance, nous avons utilisé la distance euclidienne dans une dimension 2D :

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

- Calculer la fonction de coût pour chaque particule.
 - **Cost** = $\beta * f1 + (1 - \beta) * f2$
 - **f1** = $\max_{k=1,2,\dots,k} (\sum_{ni \in C_{p,k}} d(n_i, CH_{p,k}) / |C_{p,k}|)$
 - **f2** = $(\sum_{i=1}^n E(n_i)) / (\sum_{k=1}^k E(CH_{p,k}))$

Avec :

- **β** : une constante que l'utilisateur doit fixer.
 - **f1** : la moyenne maximale de la distance euclidienne entre les nœuds et leurs CHs associés.
 - **$|C_{p,k}|$** : nombre de nœuds appartenant au cluster **C_k** de la particule **p**.
 - **f2** : le rapport entre l'énergie résiduelle totale du réseau et l'énergie résiduelle totale des CHs.
- Trouver la particule qui a la fonction de coût minimal et la conserver pour la prochaine itération.
 - A partir de la 2^{ème} itération, nous choisissons (n-1) particules en plus de la particule minimale générée à chaque fin d'itération.
 - Répéter jusqu'à stabilité du réseau : générer une particule optimale.

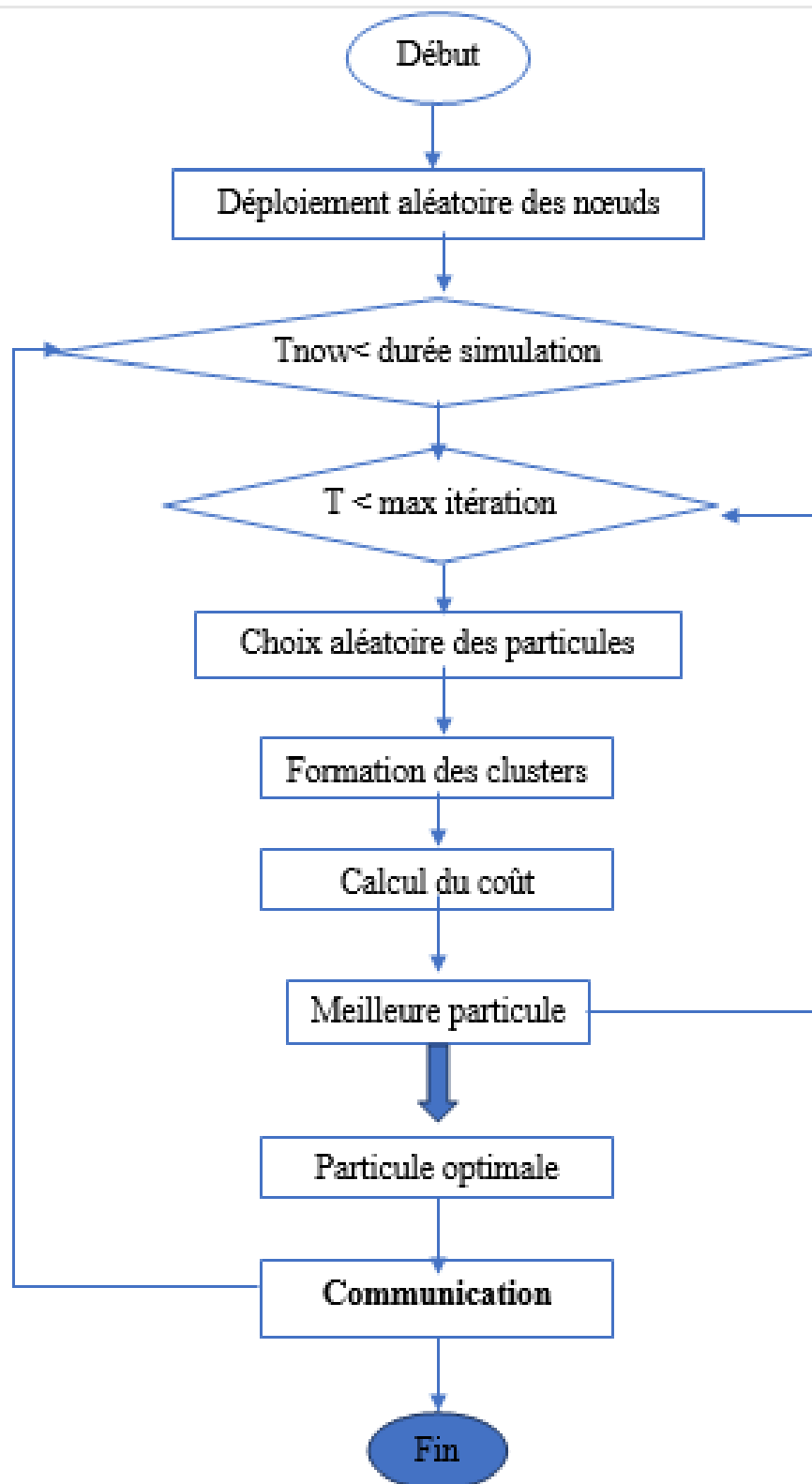


Figure 3: Organigramme du Firefly-BE.

Algorithme 1 : Firefly-BE.

Données :

- N nœuds déployés aléatoirement.
- Générer L : liste de S particules contenant K CHs choisis aléatoirement.

Résultat : Particule optimale.

1 Début

2 Tant que ($t < \text{Max itération}$)

- 3 Assigner chaque nœud n au CH le plus proche.
- 4 Calculer la fonction de coût.
- 5 Choisir la particule minimale (PM).
- 6 Ajouter PM et S-1 nouvelles particules dans L.

7 Fin Tant que

- 8 Communication intra-cluster et inter-cluster.

9 Fin

Figure 4: L'algorithme Firefly-BE.

2- Phase 2 : Communication

a- Communication intra-cluster : (TDMA)

- Un round est divisé en frame (selon le nombre de cluster).
- Dans chaque frame, nous avons des slots où les nœuds ordinaire (chacun son tour) envoient les informations captées au CH de leur cluster.
 - La communication consomme l'énergie selon le modèle suivant :

$$\begin{aligned} & \text{Transmission : } E_{TX}(l,d) = E_{TX-elect}(l,d) + E_{TX-amp}(l,d) \\ & \left\{ \begin{array}{l} l.E_{elect} + l. \epsilon_{fs}.d^2, \text{ if } d < d_0 \\ l.E_{elect} + l. \epsilon_{mp}.d^4, \text{ if } d \geq d_0 \end{array} \right. \end{aligned}$$

Où :

- l : nombre de bit de message.
 - d : la distance entre l'émetteur et le récepteur.
 - d_0 : la distance de transmission de seuil, donnée par
$$d_0 = \sqrt[4]{\epsilon_{fs} / \epsilon_{mp}}.$$
 - E_{elect} : la consommation d'énergie par bit.
 - ϵ_{fs} et ϵ_{mp} dépendent du modèle d'amplificateur de transmetteur utilisé
- Réception : $E_{RX}(l,d) = E_{elect}$

b- Schéma de mobilité de BS

- A la fin de chaque round, les CHs envoient les données reçues à la BS qui suivra le chemin déterminé par l'algorithme suivant.

2.2.2. Wolf Search Sink Moving (WS-SM)

L'algorithme de recherche des loups (WSA) est un nouvel algorithme qui s'ajoute à la famille des algorithmes d'optimisation par essaims. Cette méthode d'optimisation est utilisée récemment par des chercheurs tel que Song [21] qui a étudié le comportement de chasse de plusieurs groupes de loups. Chaque groupe de loups s'oriente selon un champ de vision déterminé, et se déplace ensuite selon la meilleure position, c'est-à-dire vers le groupe de loups qui a déjà repéré la proie.

2.2.2.1. Principe

Notre algorithme a pour but d'augmenter les performances du réseau par la minimisation de la consommation d'énergie et l'augmentation de la durée de vie des capteurs. La fonction objective est la proie la plus faible. Afin d'adapter le WSA à notre contexte, le loup est représenté ici par la BS mobile et possède un champ de vision (v) ainsi qu'une vitesse de déplacement α , les proies sont représentées par les CHs. La BS suit deux types de comportements : le comportement de chasse et la patrouille cardinale.

Le loup (BS mobile) chasse les proies qui optimisent la fonction de coût, représentée ici par le CH avec le plus de membres suivant l'un de deux comportements cités précédemment.

Les variables utilisées sont les suivantes :

- **X_i** la position actuelle du loup (x, y).
- **α** la vitesse de déplacement du loup.
- **V** le champ de vision du loup.
- **Step_size** le nombre de pas.
- **Direction** l'une des 8 directions que le loup emprunte lors de la patrouille cardinale.
- **Particule** liste des CHs du round actuel.
- **Liste_CH_visités** liste contenant la liste des CHs que la BS a déjà visités.
- **Liste_CH_pot** liste des CHs compris dans le champ de vision de la BS.

Le déroulement de cet algorithme passe par les étapes suivantes à la fin du clustering :

1- Etape 1 : Initialisation des paramètres

On définit un seul loup qui est la BS située à la position X_i ($AreaSize/2$, $AreaSize/2$), ainsi que **Particule** qui est la liste des CHs obtenue à la fin du clustering. On définit le champ de vision de la BS (200), $step_size = 1$, la vitesse de déplacement $\alpha = 0.2$ [21] et $direction = 1$. Les listes **Liste_CH_pot** et **Liste_CH_vistés** sont initialement vides.

2- Etape 2 : Scan de la région

La BS scan la région couverte par son champ de vision v afin de localiser les CHs. Si un ou plusieurs CHs sont repérés, ils sont ajoutés à la liste **Liste_CH_pot** en attente de la décision de la BS. La BS parcourt **Liste_CH_pot** et choisi le CH qui satisfait la fonction de coût et donc avec le plus grand nombre de capteurs membres et entame le **comportement de chasse** où il se dirige vers le CH avec une vitesse α selon la formule suivante :

$$X_v = X_i + Vec \times step_size \times \alpha \times rand(0,1)$$

- $X_i(x_i, y_i)$: L'emplacement actuel de la BS.
- $X_v(x_v, y_v)$: L'emplacement où la BS se déplacera.
- Vec : vecteur de translation du point X_i vers le point X_{ch} calculé comme suit :

$$Vec = X_{ch} - X_i = (x_{ch} - x_i) + (y_{ch} - y_i),$$

Où X_{ch} est la position du CH choisit.

- α : la vitesse de déplacement du loup.
- $Step_size$: le nombre de pas.
- $Rand(0,1)$: un nombre aléatoire généré entre 0 et 1.

La BS se déplace de cette manière jusqu'à atteindre une certaine distance (≤ 145) et entame la transmission de données. Une fois la transmission terminée, la BS regagne sa position initiale (à savoir X_i ($AreaSize/2$, $AreaSize/2$)), la direction est réinitialisée à 1, et on ajoute le CH visité à la liste **Liste_CH_vistés**.

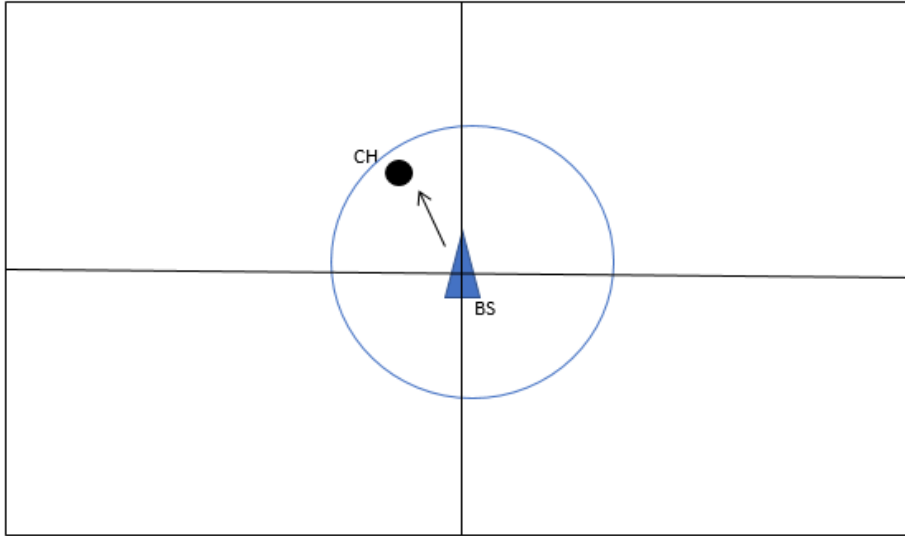


Figure 5: Représentation du comportement de chasse

3- Etape 3 : Parcours cardinal (facultatif)

Le parcours cardinal est lancé lorsqu'aucun CH n'est compris dans le champ de vision de la BS. La BS, du centre de la zone, se dirige en premier lieu vers le Nord-ouest de la zone qui correspond à direction = 1, plus précisément vers le point X_{dest} (0,0) avec la formule suivante :

$$X_v = X_i + Vec \times 0.01$$

- $X_i(x_i, y_i)$: L'emplacement actuel de la BS.
- $X_v(x_v, y_v)$: L'emplacement où la BS se déplacera.
- Vec : vecteur de translation du point X_i vers le point X_{dest} calculé comme suit :

$$Vec = X_{dest} - X_i = (x_{dest} - x_i) + (y_{dest} - y_i),$$

- 0.01 est le pas avec lequel la BS se déplace afin de pouvoir détecter tous les capteurs dans sa trajectoire.

A chaque déplacement, la BS procède à un Scan de la région (**Etape 2**) et suit le déroulement de cette dernière. Par contre, si la BS atteint sa destination (le point X_{dest}), elle regagne le centre de la zone, incrémente la variable direction pour répéter l'étape 3 pour la prochaine destination.

Les huit destinations (représentés par les points cardinaux) sont les points suivants de la zone :

- Destination = 1 : $X_{\text{dest}}(x_{\text{dest}}, y_{\text{dest}}) = (0,0)$. //Nord-ouest
- Destination = 2 : $X_{\text{dest}}(x_{\text{dest}}, y_{\text{dest}}) = (\text{AreaSize}/2, 0)$. //Nord
- Destination = 3 : $X_{\text{dest}}(x_{\text{dest}}, y_{\text{dest}}) = (\text{AreaSize}, 0)$. //Nord-est
- Destination = 4 : $X_{\text{dest}}(x_{\text{dest}}, y_{\text{dest}}) = (\text{AreaSize}, \text{AreaSize}/2)$. //Est
- Destination = 5 : $X_{\text{dest}}(x_{\text{dest}}, y_{\text{dest}}) = (\text{AreaSize}, \text{AreaSize})$. //Sud-est
- Destination = 6 : $X_{\text{dest}}(x_{\text{dest}}, y_{\text{dest}}) = (\text{AreaSize}/2, \text{AreaSize}/2)$. //Sud
- Destination = 7 : $X_{\text{dest}}(x_{\text{dest}}, y_{\text{dest}}) = (0, \text{AreaSize})$. //Sud-ouest
- Destination = 8 : $X_{\text{dest}}(x_{\text{dest}}, y_{\text{dest}}) = (0, \text{AreaSize}/2)$. //Ouest

Parcourir la zone de cette manière garantie que la BS balaye toute la zone pour repérer tous les CHs non visités.

On répète les étapes 2 et 3 jusqu'à ce que la taille de la liste **Liste_CH_visités** soit égale à la taille de la liste **Particule**, c'est-à-dire jusqu'à ce que tous les CHs du round sont visités et transmis leurs données à la BS.

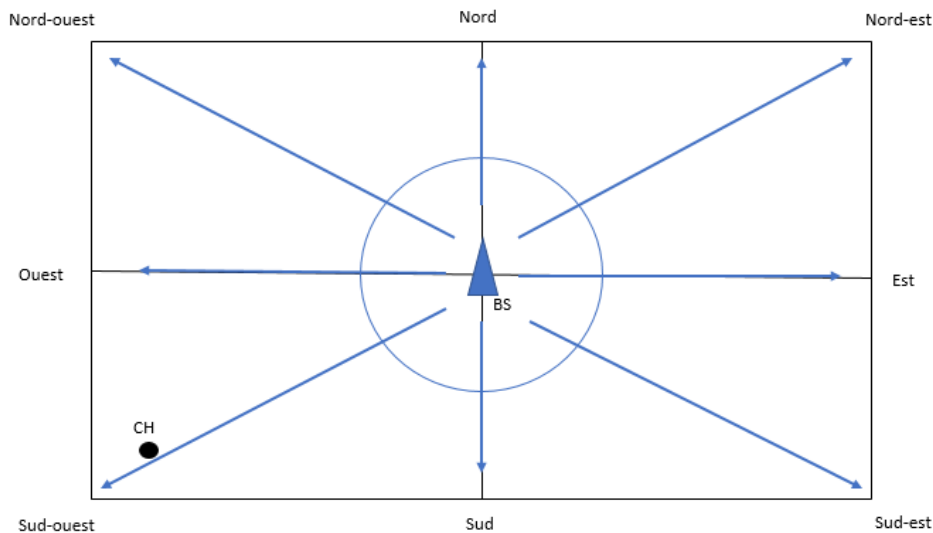


Figure 6: Représentation du parcours cardinal.

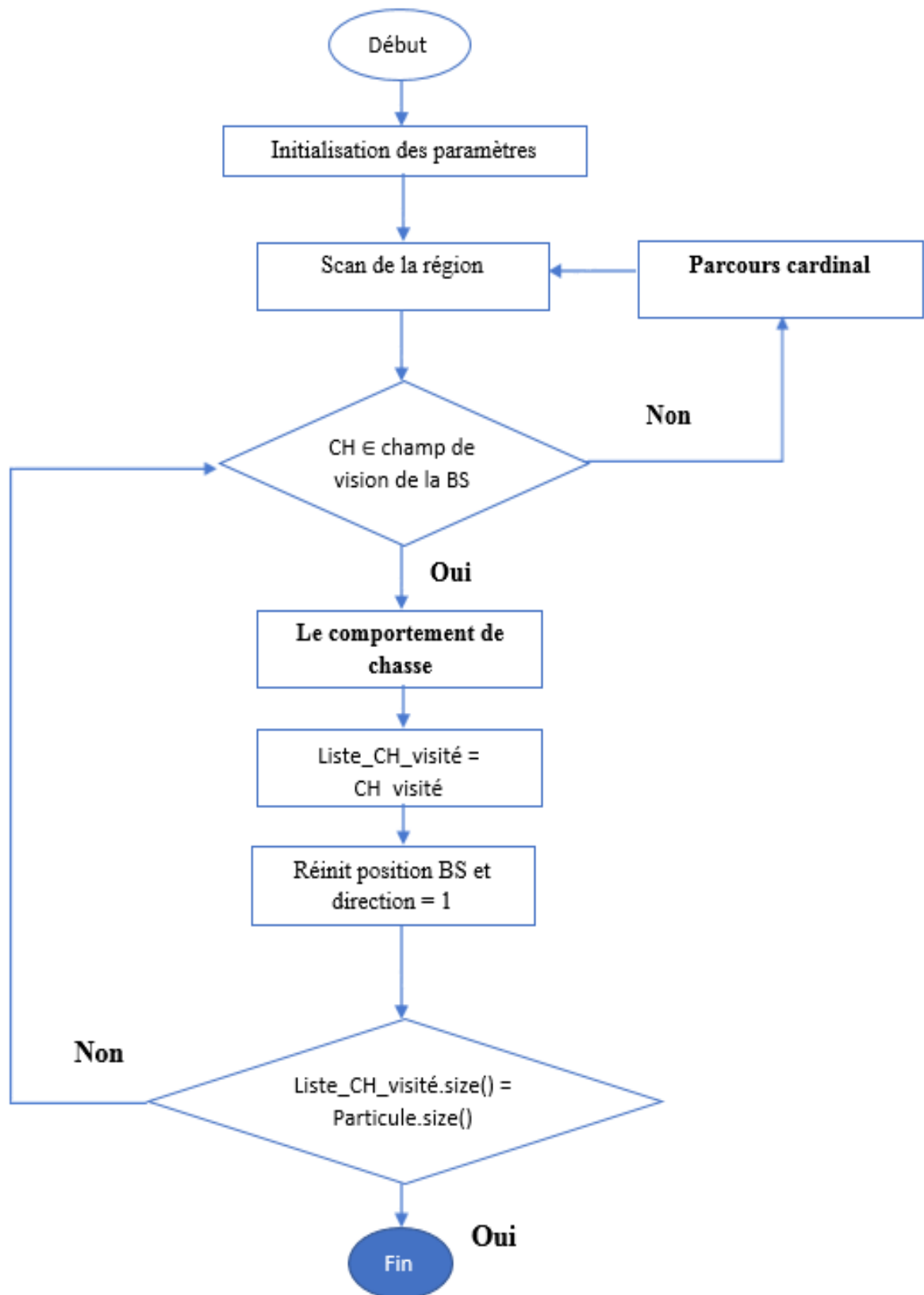


Figure 7: Organigramme du WS-MS.

Algorithme 2: Wolf Search Moving Sink

Données :

- P : particule de CHs résultante de l'algorithme Firefly.
- M : liste des CHs visités par la BS mobile initialement vide.

Résultat : Taille de M = Taille de Particule.

1 Début

2 **Tant que** (Taille M \neq Taille Particule)

3 **Scan** de la région.

4 **Si** (CH \in champ de vision de la BS)

5 Comportement de chasse.

6 Liste_CH_visit   = CH_visit  .

7 R  nit position BS et direction = 1.

8 **Sinon**

9 Parcours cardinal.

10 Aller a 3.

11 **Fin Si**

12 **Fin Tant que**

13 Fin

Figure 8: L'algorithme WS-MS.

- Resultats de la simulation

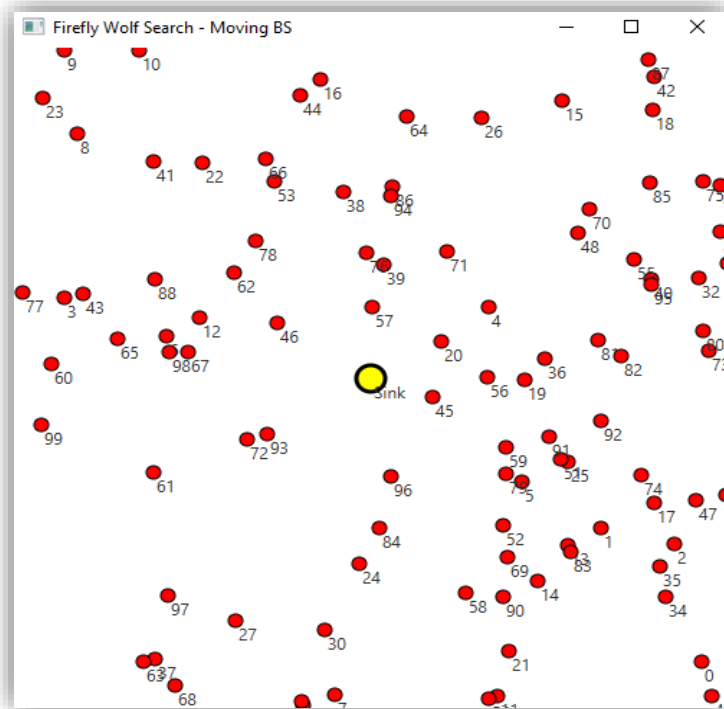


Figure 9: Capture d'écran lors de l'initialisation du réseau.

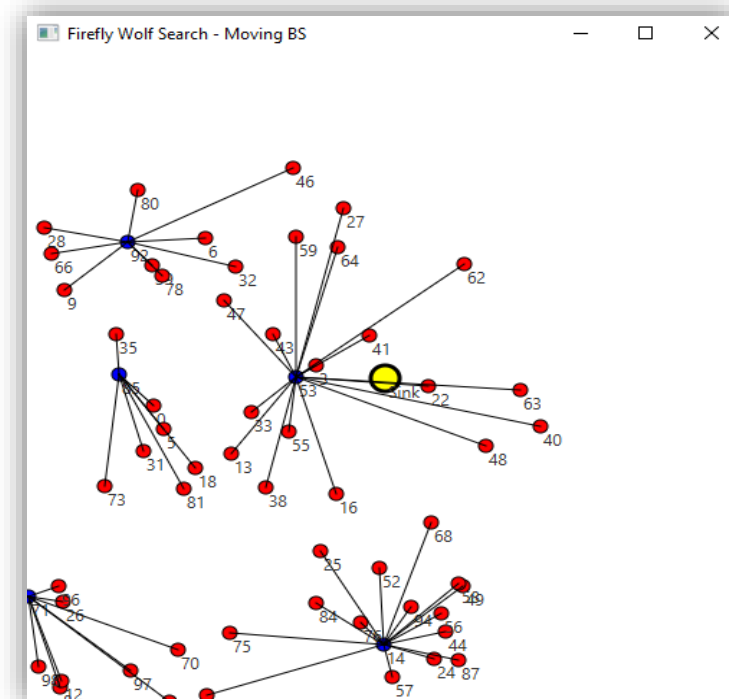


Figure 10: Capture d'écran lors de l'instant $t = 1000s$.

2.3.Conclusion

Dans ce chapitre, nous avons présenté notre protocole FFWS-MS en nous inspirant de l'algorithme de recherche des loups comme approche bio-inspiré. Nous avons aussi proposé un algorithme de clustering basé sur l'algorithme des lucioles. Nous avons expliqué le modèle de mobilité de la BS, selon le comportement d'un loup. La combinaison de ces deux algorithmes fait que l'on optimise considérablement la quantité d'énergie consommée, améliore la durée de vie du RCSF et augmente le nombre de data reçues par la BS.

3.1. Introduction

Dans ce chapitre, nous allons évaluer les performances de nos schémas proposés en comparaison avec l'algorithme *gbest* PSO [22].

Nous avons implémenté nos schémas sous le langage Java, où nous avons développé notre propre simulateur à l'aide de la technologie Java-FX.

3.2. L'environnement de développement

Java-FX est une bibliothèque très riche graphiquement qui permet de réaliser des animations assez poussées. Notre choix s'est porté sur cet environnement car il nous permet une certaine liberté du fait que l'on implémente notre propre simulateur, mais aussi la possibilité d'obtenir un rendu graphique clair et net, ajouté à cela la possibilité de dessiner différents graphes et courbes pour visualiser les résultats de notre simulation.

3.3. Simulation et évaluation

3.3.1. Les paramètres de simulation

Tableau 1: Tableau des différents paramètres de simulation.

Paramètres	Valeurs
Espace	100*100 – 500*500 m ²
Nœuds	100 – 200
Nombre de particules	10% des nœuds
Nombre de CHs	5% des nœuds
Energie initial	2 J
E_{elect}	50nJ/bit
ϵ_{fs}	10pJ/bit/m ²
ϵ_{mp}	0.0013pJ/bit/m ²
La taille du message	4000 bits
Nombre d'itération	100
TDMA (Round)	20 – 30 ms
Slot	0.05 sec

3.3.2. Comparaison des performances

Les principaux paramètres d'évaluation d'un schéma de routage sont :

- Les nœuds survivants.
- L'énergie consommée.
- Le nombre total de données envoyées par les nœuds et les CHs.

Afin d'évaluer les performances de nos propositions, nous avons fait une étude comparative entre la combinaison de nos deux contributions et le schéma bio-inspiré Partical Swarm optimisation (PSO). Notons que pour chaque série de simulations, différentes topologies sont testées en modifiant la taille du réseau, la durée du round ainsi que la surface. Ces critères sont résumés dans la figure suivante :

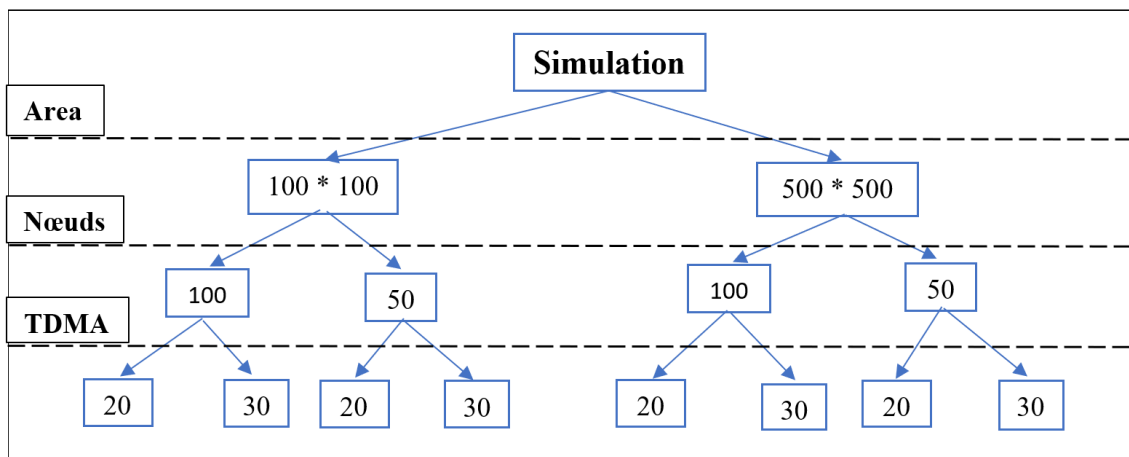


Figure 11: Les divers cas de notre étude.

Au cours de nos simulations, tous les nœuds sont homogènes. Quelques paramètres sont fixés dont le plus important est : Le déploiement aléatoire des nœuds. La durée de simulation n'est pas déterminée et se poursuit jusqu'à impossibilité de formation de cluster. La position de la BS est toujours fixée à $(AreaSize/2 \times AreaSize/2)$ et possède une énergie illimitée. Le tableau suivant montre les valeurs initialisées des paramètres des algorithmes comparés :

Tableau 2 : Table des paramètres.

Paramètres	Valeurs
α	0.2
Step_size	1
Champ de vision	200 m
β	0.2
C_1 et C_2	2
W_{min}	0.9
W_{max}	0.4

Pour tracer les graphes de ces algorithmes, nous avons utilisé des programmes écrits en Java-FX et les résultats sont représentés par les graphes suivants :

1- Partie 1 : Comparaison dans un réseau de petite taille (100 * 100)

a- Simulation avec 100 nœuds

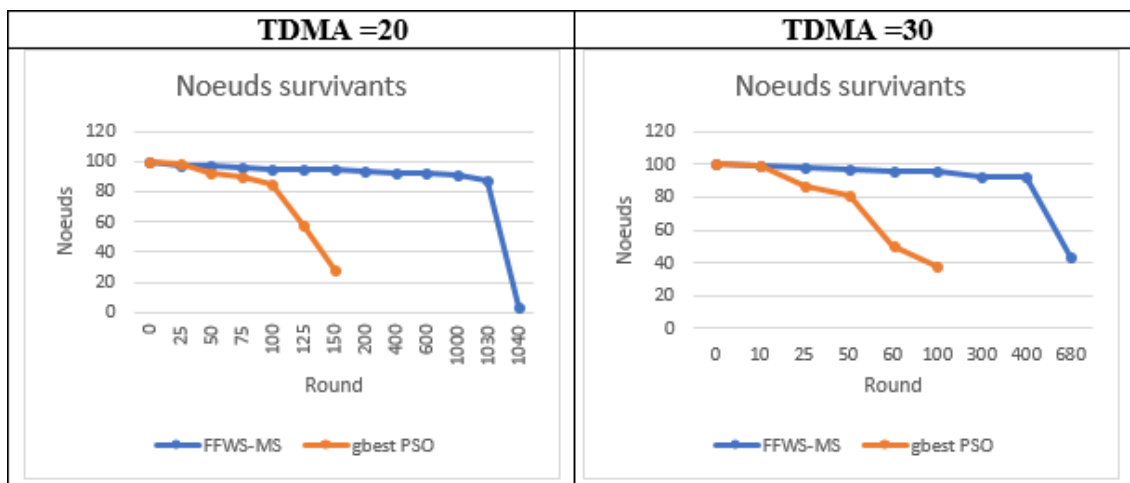


Figure 12: Nombre de nœuds survivants en *gbest* PSO et FFWS-MS (A= 100*100 et N = 100).

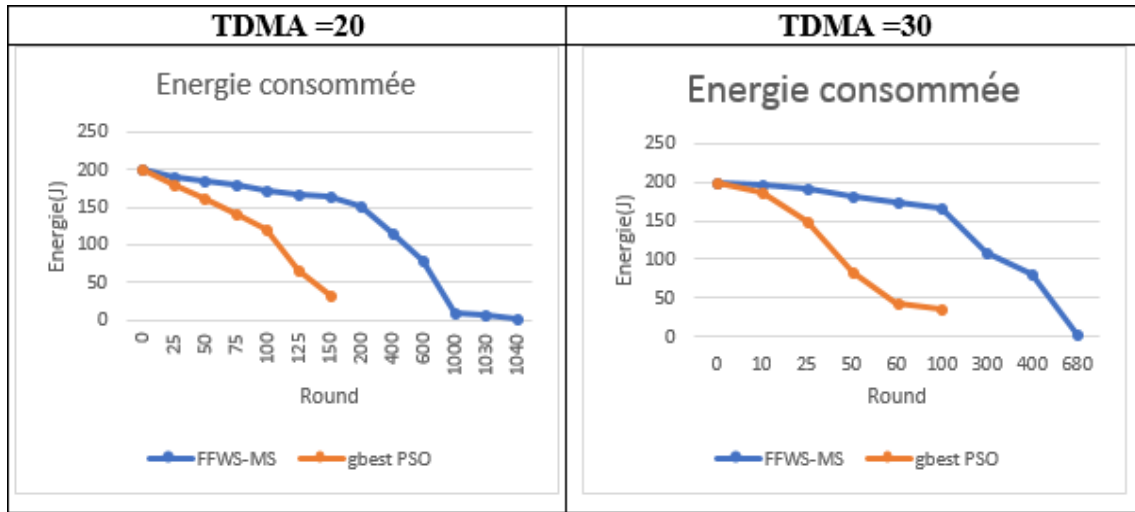


Figure 14: Energie consommée en *gbest* PSO et FFWS-MS (A= 100*100 et N = 100).

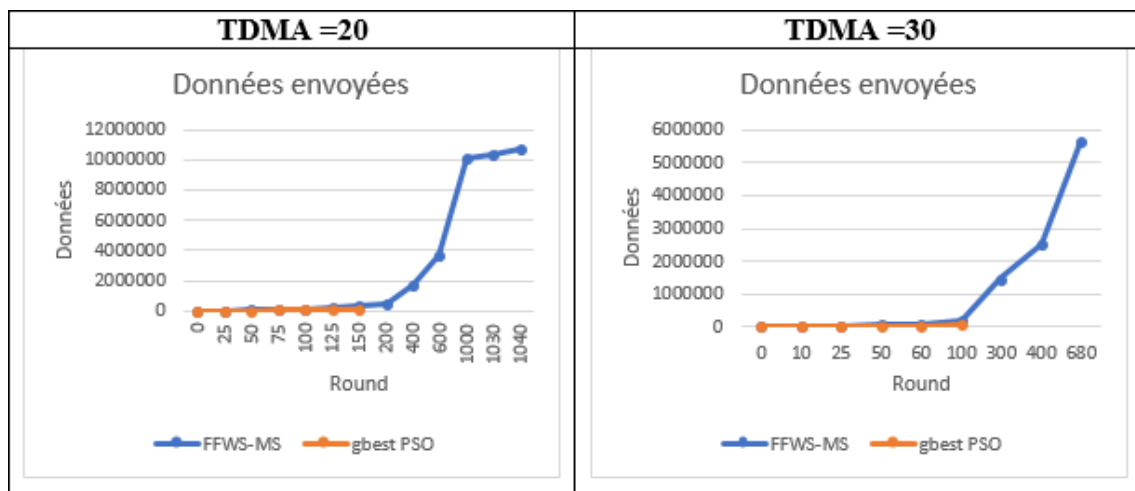


Figure 13: Nombre de données envoyées en *gbest* PSO et FFWS-MS (A= 100*100 et N = 100).

Discussion

a) Comparaison de la durée de vie

La figure 12 représente la durée de vie du réseau durant la simulation du schéma *gbest* PSO et notre contribution le FFWS-MS selon deux cas différents :

- D'après la figure 12 où la durée du round est 20 ms, on remarque que le FFWS-MS assure une longue durée de vie du réseau et cela jusqu'au TDMA (1040). Il permet de maintenir une longue stabilité de survie de nœuds jusqu'au TDMA (1030), tandis que pour le *gbest* PSO on remarque que le nombre de nœuds diminue considérablement à partir du TDMA (100) et ne dépasse pas le TDMA (150).
- En fixant la durée du round à 30 ms, on remarque que les résultats du *gbest* PSO sont quasi similaires à la précédente étude, tandis que le FFWS-MS garde toujours une stabilité du nombre de nœuds survivants mais décroît considérablement au TDMA (680).

b) Comparaison de l'énergie consommée

La figure 13 représente la quantité d'énergie consommée par les nœuds durant la simulation du *gbest* PSO et de notre contribution le FFWS-MS :

- D'après la figure 13, on remarque une grande différence de consommation d'énergie entre les deux schémas. Pour ce qui est du *gbest* PSO, l'ensemble des capteurs épuisent leurs énergies au TDMA (150). Quant au FFWS-MS, il ne consomme cette énergie qu'au TDMA (1040) grâce au choix des CHs optimaux obtenus lors du clustering mais aussi à la distance réduite entre les CHs et la BS lors de son déplacement.
- En fixant le round à 30 ms, on remarque que pour *gbest* PSO les résultats sont quasi similaires à la précédente étude où le réseau se déconnecte lors du TDMA (100), tandis que pour le FFWS-MS, l'ensemble des nœuds épuisent leurs énergies au TDMA (680). Cela est due à l'augmentation de la durée du round où l'envoi intra-cluster croît considérablement.

c) Comparaison des données envoyées à la BS

La figure 14 représente le nombre de données envoyées à la BS durant la simulation du *gbest* PSO et FFWS-MS :

- La figure 14 indique une large différence entre le nombre de données envoyées à la BS lors de l'étude entre les deux schémas. L'ensemble des données envoyées lors de l'utilisation du *gbest* PSO (57160 messages) au TDMA (150) et grandement négligeable face au nombre de messages envoyés lors de l'utilisation du FFWS-MS (près de 10795342 messages) au TDMA (1040).
- Les résultats montrés lorsque l'on fixe le TDMA à 30 ms sont similaires du point de vue du grand écart présent entre le nombre total de données envoyées à la BS. Néanmoins, la différence se trouve dans le nombre total de rounds où l'ensemble des données envoyées lors de l'utilisation du *gbest* PSO est de 57108 messages au TDMA (150), tandis que lors de l'utilisation du FFWS-MS la transmission s'arrête au TDMA (680) avec un total de 5637907 messages.

b- Simulation avec 200 nœuds

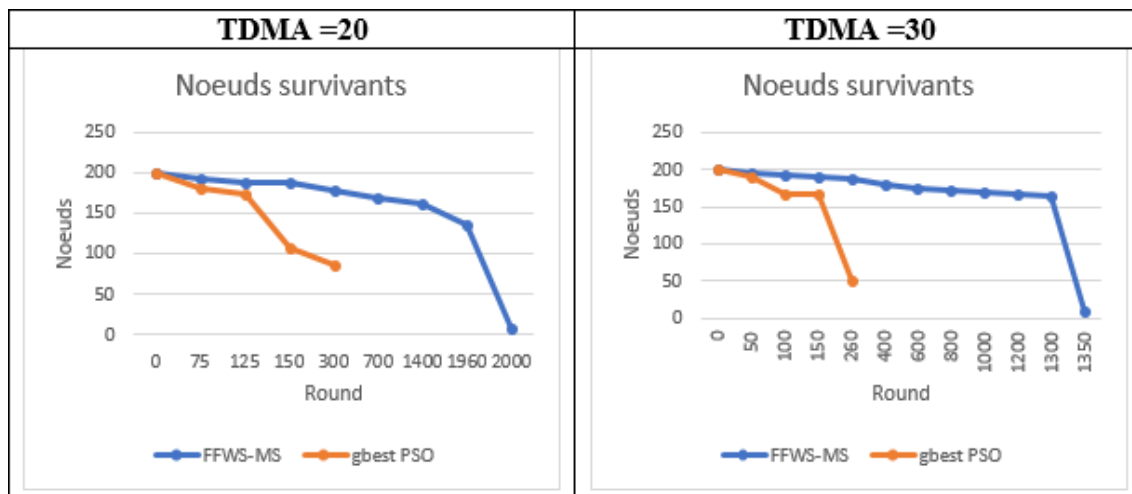


Figure 15: Nombre de nœuds survivants en *gbest* PSO et FFWS-MS (A= 100*100 et N = 200).

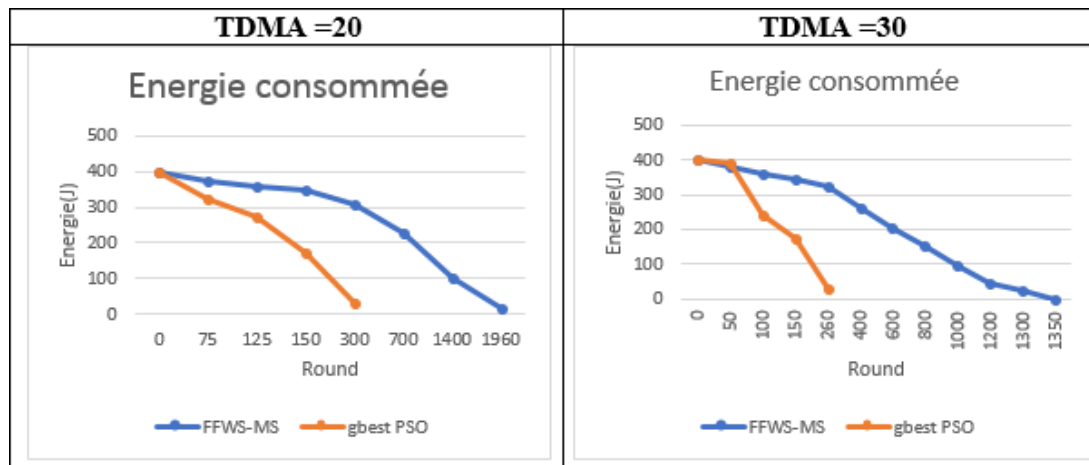


Figure 16: Energie consommée en *gbest* PSO et FFWS-MS ($A=100 \times 100$ et $N=200$).

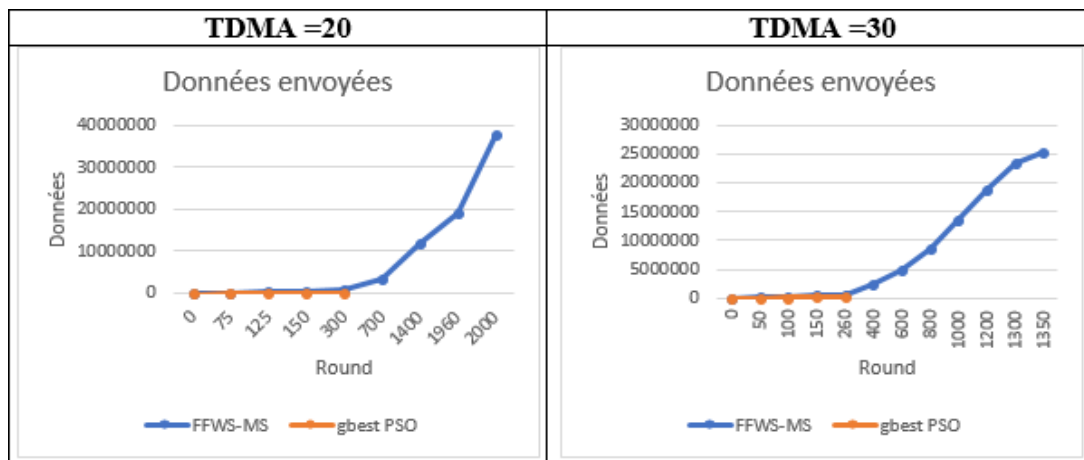


Figure 17: Nombre de données envoyées en *gbest* PSO et FFWS-MS ($A=100 \times 100$ et $N=200$).

Discussion

a) Comparaison de la durée de vie

La figure 15 représente la durée de vie du réseau durant la simulation du *gbest* PSO et notre contribution le FFWS-MS :

- D'après la figure 15, on remarque que le FFWS-MS assure une longue durée de vie du réseau et cela jusqu'au TDMA (2000) et qu'il permet de maintenir une longue stabilité de survie de nœuds jusqu'au TDMA (1960), tandis que pour le *gbest* PSO on remarque que le nombre de nœuds commence à se réduire considérablement à partir du TDMA (125) et ne dépasse pas le TDMA (300). On déduit que notre schéma améliore la durée de vie du réseau à grande échelle.
- En fixant le round à 30 ms, on remarque que les résultats du *gbest* PSO sont quasi similaires à la précédente étude, tandis que le FFWS-MS garde toujours une stabilité du nombre de nœuds survivants mais décroît considérablement au TDMA (1300).

b) Comparaison de l'énergie consommée

La figure 16 représente la quantité d'énergie consommée par les nœuds durant la simulation du *gbest* PSO et notre contribution le FFWS-MS :

- D'après la figure 16, on remarque une grande différence de consommation d'énergie entre les deux schémas. Pour ce qui est du *gbest* PSO, l'ensemble des capteurs épuisent leurs énergies au TDMA (300). Quant au FFWS-MS, il ne consomme cette énergie qu'au TDMA (2000) grâce aux choix de CHs optimaux obtenus lors du clustering mais aussi à la distance réduite entre les CHs et la BS lors de son déplacement.
- En fixant le round à 30 ms, on remarque que pour *gbest* PSO, les résultats sont quasi similaires à la précédente étude où le réseau se déconnecte lors du TDMA (260), tandis que pour le FFWS-MS, l'ensemble des nœuds

épuisent leurs énergies au TDMA (1350). Cela est dû à l'augmentation de la durée du round où l'envoi intra-cluster croît considérablement.

c) Comparaison des données envoyées à la BS

La figure 17 représente le nombre de données envoyées à la BS durant la simulation du *gbest* PSO et notre contribution le FFWS-MS :

- La figure 17 indique une large différence entre le nombre de données envoyées à la BS lors de l'étude entre les deux schémas. L'ensemble des données envoyées lors de l'utilisation du *gbest* PSO (116691) au TDMA (300) est grandement négligeable face au nombre de messages envoyées lors de l'utilisation du FFWS-MS (près de 37905274) au TDMA (2000).
- Les résultats montrés par la figure 17 lorsque l'on fixe le TDMA à 30 sont similaires du point de vue du grand écart présent entre le nombre total de données envoyées à la BS. Néanmoins, la différence se trouve dans le nombre de rounds total où l'ensemble des données envoyées lors de l'utilisation du *gbest* PSO est de 49607 au TDMA (360), tandis que lors de l'utilisation du FFWS-MS la transmission s'arrête au TDMA (1350) avec un total de 25119031 messages.

2- Partie 2 : comparaison dans un réseau de grande taille (500 * 500)

a- Simulation avec 100 nœuds

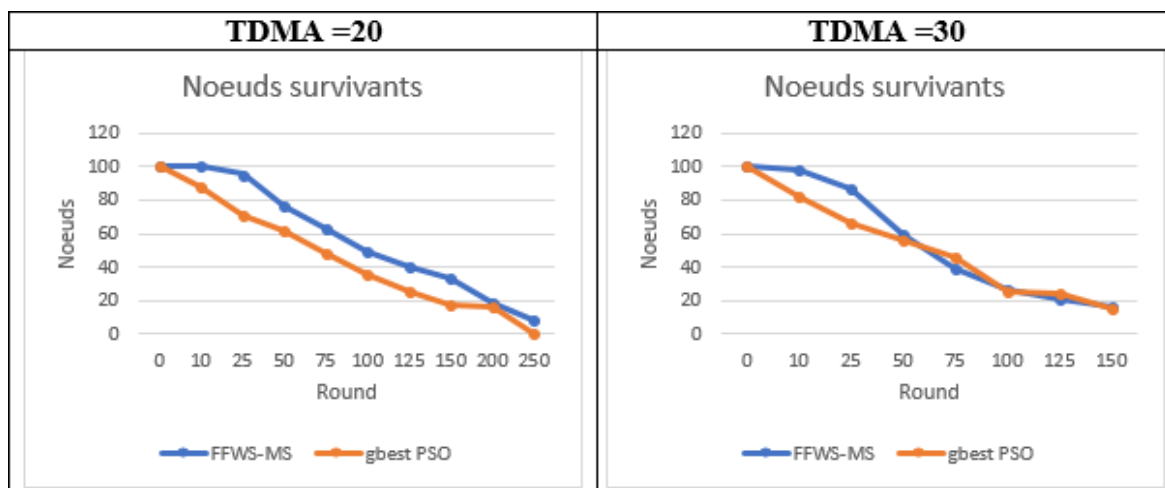


Figure 18: Nombre de nœuds survivants en *gbest* PSO et FFWS-MS ($A = 500 * 500$ et $N = 100$).

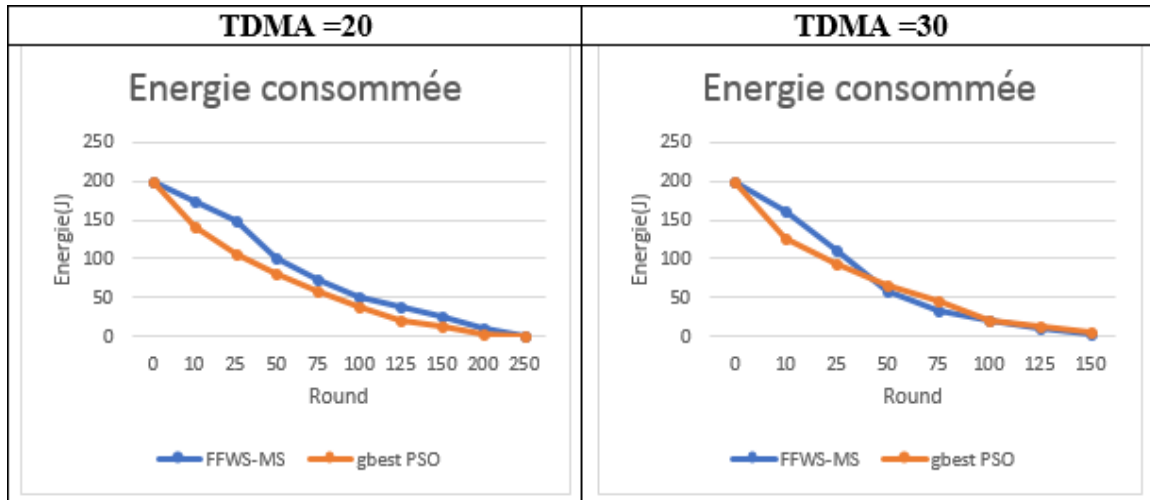


Figure 19: Energie consommée en *gbest* PSO et FFWS-MS ($A = 500 * 500$ et $N = 100$).

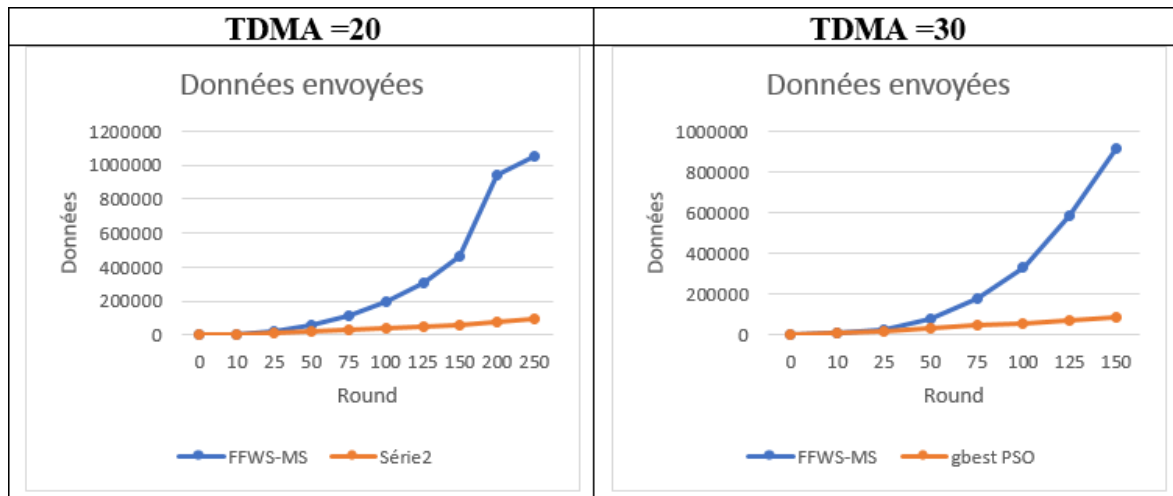


Figure 20: Nombre de données envoyées en *gbest* PSO et FFWS-MS ($A = 500 * 500$ et $N = 100$).

Discussion

a) Comparaison de la durée de vie

La figure 18 représente la durée de vie du réseau durant la simulation du schéma *gbest* PSO et notre contribution le FFWS-MS selon deux cas différents :

- D'après la figure 18, on remarque que le *gbest* PSO garde les nœuds vivants jusqu'au TDMA (250), tandis que le FFWS-MS procure une durée de vie du réseau assez courte (similaire au *gbest* PSO). Cela est dû aux longues distances qui séparent les nœuds sur une plus grande superficie, ce qui consomme plus d'énergie lors de la communication intra-cluster.
- En fixant le TDMA à 30, on remarque que les résultats de cette simulation réduisent la durée de vie du réseau pour les deux schémas jusqu'au TDMA (150). Cela revient aux multiples envois intra-cluster coûteux en énergie dans une plus longue durée de round.

b) Comparaison de l'énergie consommée

La figure 19 représente la quantité d'énergie consommée par les nœuds durant la simulation du schéma *gbest* PSO et notre contribution le FFWS-MS selon deux cas différents :

- D'après la figure 19, on remarque des résultats quasi similaires de consommation d'énergie entre les deux schémas. Pour ce qui est du *gbest* PSO, l'ensemble des capteurs épuisent leurs énergies au TDMA (250). Quant au FFWS-MS, il consomme également toute cette énergie au TDMA (250).
- En fixant le TDMA à 30, on remarque dans la figure 19 que les résultats sont proches des résultats de la simulation précédente sauf que pour les deux schémas, l'énergie des capteurs s'épuise au TDMA (150).

c) Comparaison des données envoyées à la BS

La figure 20 représente le nombre de données envoyées à la BS durant la simulation du schéma *gbest* PSO et notre contribution le FFWS-MS :

- La figure 20 indique que l'ensemble des données envoyées lors de l'utilisation du *gbest* PSO est de 94824 au TDMA (250), tandis que lors de l'utilisation du FFWS-MS, les résultats sont de 1053885 messages envoyés au TDMA (250).
- Les résultats montrés par la figure 20 indiquent que l'ensemble des données envoyées lors de l'utilisation du *gbest* PSO est de 89131 au TDMA (150), tandis que lors de l'utilisation du FFWS-MS, les résultats sont de 913870 messages envoyés au TDMA (150).

a- Simulation avec 200 nœuds

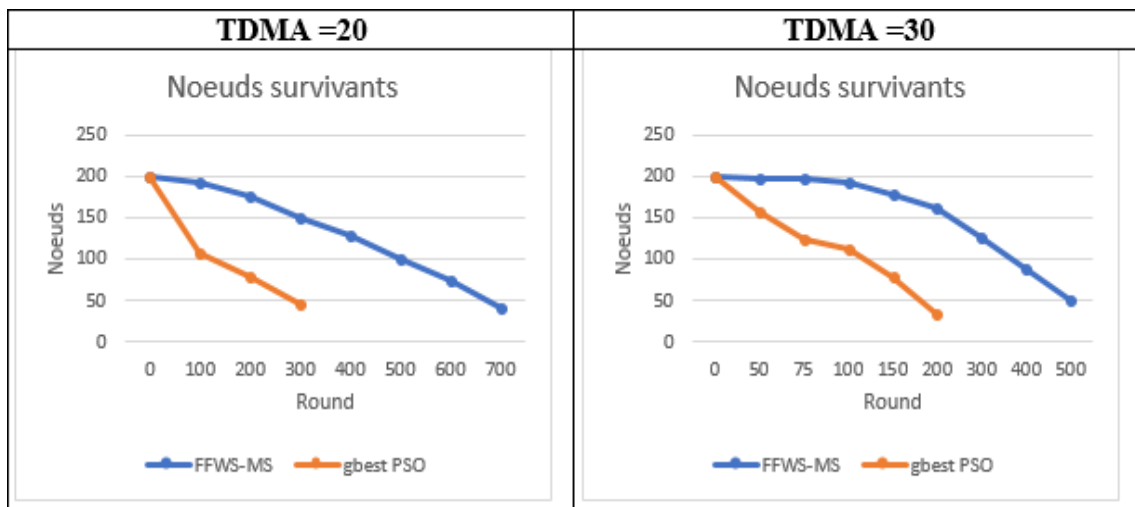


Figure 21: Nombre de nœuds survivants en *gbest* PSO et FFWS-MS (A= 500 * 500 et N = 200).

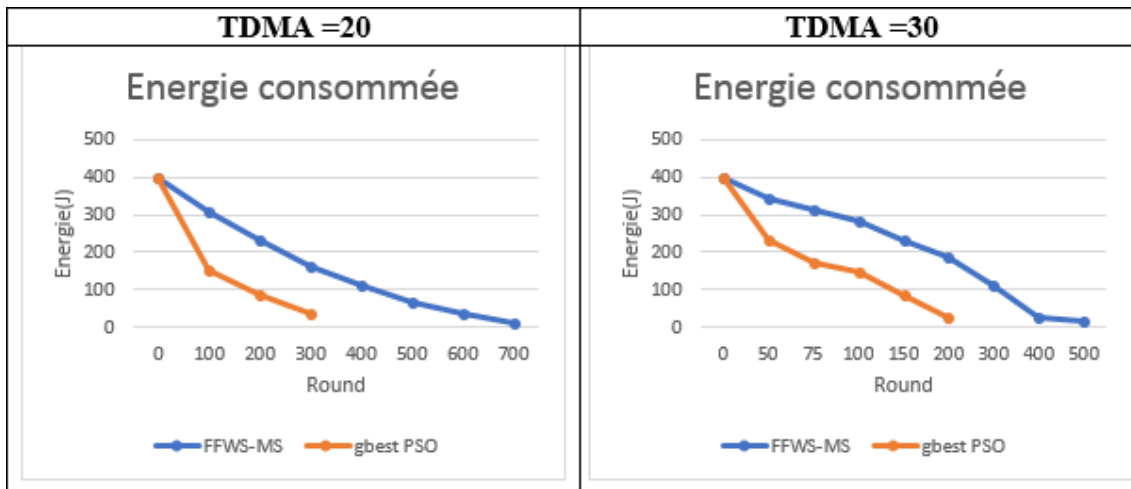


Figure 22: Energie consommée en *gbest* PSO et FFWS-MS ($A= 500 * 500$ et $N = 200$).

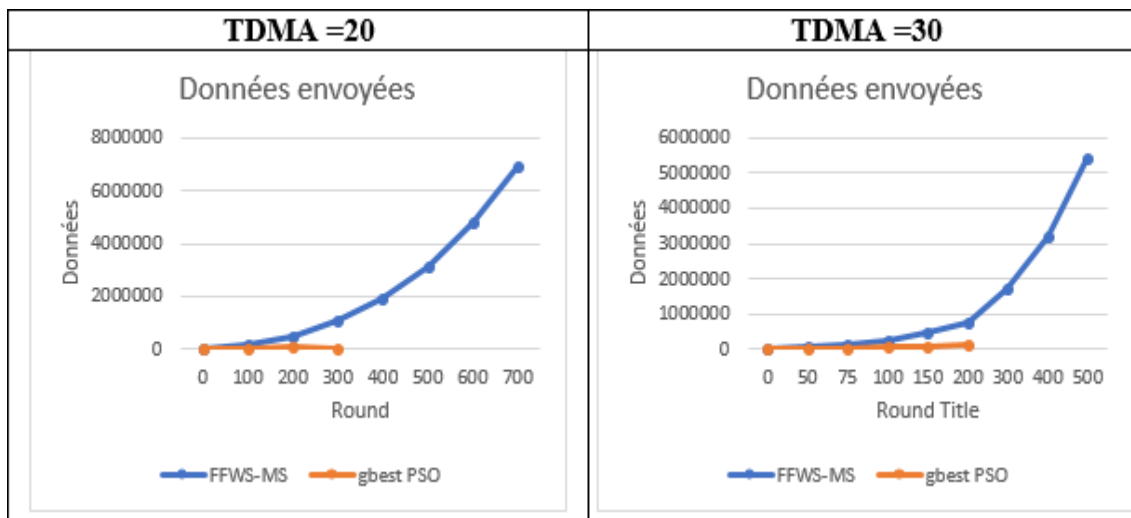


Figure 23: Nombre de données envoyées en *gbest* PSO et FFWS-MS ($A=500* 500$ et $N=200$).

Discussion

a) Comparaison de la durée de vie

La figure 21 représente la durée de vie du réseau durant la simulation du schéma *gbest* PSO et notre contribution le FFWS-MS :

- D'après la figure 21, on remarque que le FFWS-MS assure une longue durée de vie du réseau et cela jusqu'au TDMA (700), tandis que pour le *gbest* PSO on remarque que la durée de vie du réseau ne dépasse pas le TDMA (300).
- En fixant le TDMA à 30, on remarque dans la figure que le FFWS-MS assure toujours de meilleures performances mais uniquement jusqu'au TDMA (500), alors que le *gbest* PSO permet une durée de vie du réseau jusqu'au TDMA (200).

b) Comparaison de l'énergie consommée

La figure 22 représente la quantité d'énergie consommée par les nœuds durant la simulation du schéma *gbest* PSO et notre contribution le FFWS-MS :

- D'après la figure 22, on remarque une grande différence de consommation d'énergie entre les deux schémas. Pour ce qui est du *gbest* PSO, l'ensemble des capteurs épuisent leurs énergies au TDMA (300). Quant au FFWS-MS, il ne consomme cette énergie qu'au TDMA (700).
- En fixant le TDMA à 30, on remarque dans la figure 22 que le réseau se déconnecte lors du TDMA (200) pour le *gbest* PSO, tandis que pour le FFWS-MS, l'ensemble des nœuds épuisent leurs énergies au TDMA (500).

c) Comparaison des données envoyées à la BS

La figure 23 représente le nombre de données envoyées à la BS durant la simulation du schéma *gbest* PSO et notre contribution le FFWS-MS selon deux cas différents :

- La figure 23 indique une large différence entre le nombre de données envoyée à la BS lors de l'étude entre les deux schémas. L'ensemble des données envoyées lors de l'utilisation du *gbest* PSO (59449) au TDMA (300) et grandement négligeable

face au nombre de messages envoyés lors de l'utilisation du FFWS-MS (près de 6888807) au TDMA (700).

- Les résultats montrés dans la figure 23 lorsque l'on fixe le TDMA à 30 sont similaires du point de vue du grand écart présent entre le nombre total de données envoyées à la BS. Néanmoins, la différence se trouve dans le nombre rounds total où l'ensemble des données envoyées lors de l'utilisation du *gbest* PSO est de 118384 au TDMA (200), tandis que lors de l'utilisation du FFWS-MS la transmission s'arrête au TDMA (500) avec un total de 5416578 messages.

3.4. Conclusion

Dans ce chapitre, nous avons présenté et discuté les résultats de nos simulations en comparant notre schéma proposé (FFWS-MS) avec le schéma *gbest* PSO. Les résultats montrent clairement que notre proposition est plus performante que le *gbest* PSO. L'utilisation du Firefly-BE lors du clustering ainsi que la mobilité de la BS suivant l'algorithme WS-MS assurent une amélioration des performances du réseau, réduisent la consommation d'énergie et augmentent la durée de vie du réseau. FFWS-MS est plus efficace pour des réseaux à grandes échelles dans des zones relativement petites.

Conclusion générale

Dans ce travail, nous avons traité l'un des problèmes majeurs des RCSFs qui est le problème de consommation d'énergie. Nous avons proposé un nouveau schéma de mobilité de BS afin d'améliorer la durée de vie du réseau.

Les contributions apportées par ce travail traitent en premier lieu la formation optimale de clusters, inspirée de l'algorithme d'optimisation des lucioles qui vise à optimiser la consommation d'énergie des communications intra-cluster. Ensuite, nous nous sommes inspirés de l'algorithme bio-inspiré de recherche des loups afin de déplacer la station de base vers les CHs élus du round actuel pour but de réduire la distance entre la BS et le CH et pour diminuer ainsi la quantité d'énergie dépensée lors de la communication inter-cluster.

Afin de valider l'efficacité apportée par notre schéma, nous avons développé notre propre simulateur sous l'environnement Java assisté par la technologie Java-FX, où nous avons réalisé plusieurs simulations comparatives entre FFWS-MS et *gbest* PSO. Les résultats de simulation ont montré une diminution de consommation d'énergie et une maximisation de la durée de vie du réseau apportée par FFWS-MS par rapport au *gbest* PSO, ce qui approuve le rendement que la mobilité de la BS apporte à la résolution de la problématique de consommation d'énergie des RCSFs.

Comme perspectives, nous proposons les ajustements suivants afin d'améliorer nos résultats. En améliorant le choix de CHs lors de la phase de clusterisation, en cherchant une meilleure paterne de déplacement de la BS, mais aussi en un temps de séjour de la BS dans une zone lors de la rencontre de plusieurs CHs.

Bibliographie

- [1] B. Leandro Buss, S.Ali, “A Survey on Data Collection in Mobile Wireless Sensor Networks (MRCSEs)”, In “Cooperative Robots and Sensor Networks 2015”, K. Anis, J.R Martínez Dios, Switzerland, Vol. 604, 19 May 2015, pp. 257-278.
- [2] J. Long, G. Wei-hua, 2012, “Node deployment strategy optimization for wireless sensor network with mobile base station “, February 2012, Vol. 19, Issue 2, pp 453-458.
- [3] Monique Becker, André-Luc Beylot, Riadh Dhaou, Ashish Gupta, Rahim Kacimi, and Michel Marot. Experimental study: Link quality and deployment issues in wireless sensor networks. In Proceedings of the 8th International IFIP-TC6 Networking Conference (NETWORKING'09), volume 5550, pages 14-25, Aachen, Germany, 2009. Springer.
- [4] Arnab Chakrabarti, Ashutosh Sabharwal, and Behnaam Aazhang. Using predictable observer mobility for power efficient design of sensor networks. In Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03), pages 129-145, Palo Alto, CA, USA, 2003.
- [5] Enrico Natalizio, Valeria Loscri. Controlled Mobility in Mobile Sensor Networks: Advantages, Issues and Challenges, France, Italy, 2011.
- [6] R. C. Shah, S. Roy, S. Jain, and W. Brunette, “Data MULEs: modeling a three tier architecture for sparse sensor networks”, in Proc. 1st IEEE Int. Workshop Sensor Network Protocols and Applications, Alaska, pp. 30- 41, May. 2003.
- [7] M. Ma, Y. Y. Yang, and M. Zhao, “Tour Planning for Mobile Data- Gathering Mechanisms in Wireless Sensor Networks”, IEEE Trans. Veh. Technol., vol. 62, no. 4, pp. 1472-1483, Nov. 2013.
- [8] H. Nakayama and Z. M. Fadlullah, and N. Ansari, “A Novel Scheme for WSN Sink Mobility Based on Clustering and Set Packing Techniques”, IEEE Trans. Autom. Control, vol. 56, no. 10, pp. 2381- 2389, Aug. 2011.

- [9] C. Konstantopoulos, G. Pantziou, and D. Gavalas, "A Rendezvous- Based Approach Enabling Energy-Efficient Sensory Data Collection with Mobile Sinks", IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 5, pp. 809-817, Sept. 2012.
- [10] Y. Gu, Y. S. Ji, and J. Li, "ESWC: Efficient Scheduling for the Mobile Sink in Wireless Sensor Networks with Delay Constraint", IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 7, pp. 1310-1320, Jul. 2013.
- [11] J. Wang, Y. Yin, J. Zhang, S. Lee, and R. S. Sherratt, "Mobility based energy efficient and multi-sink algorithm for consumer home networks", IEEE Trans. Consum. Electron., vol. 59, no. 1, pp.77-84, Feb. 2013.
- [12] J. Wang, Y. Yin, J. Zhang, S. Lee, and R. S. Sherratt, "Bio-inspired Ant Colony Optimization based Clustering Algorithm with Mobile Sinks for Applications in Consumer Home Automation Networks", IEEE Transactions on Consumer Electronics, Vol. 61, No. 4, Nov. 2015.
- [13] Sumit Kataria and Aarti Jain. Bio Inspired Optimal Relocation of Mobile Sink Nodes in Wireless Sensor Networks. Department of Electronics and Communication Ambedkar Institute of Advanced Communication Tech. & Research, Govt. of N.C.T. of Delhi, Delhi.
- [14] Abdul Latiff, Nurul Adilah & Abdul Latiff, Nurul Muazzah & Ahmad, R.Badlishah. (2011). Prolonging lifetime of wireless sensor networks with mobile base station using particle swarm optimization. International Review on Computers and Software (IRECOS). 10. 1 - 6. 10.1109/ICMSAO.2011.5775487.
- [15] W. Lun Chang, D. Zeng, R. Chen *and al.*," *An artificial bee colony algorithm for data collection path planning in sparse wireless sensor networks*", International Journal of Machine Learning and Cybernetics, June 2015, Volume 6, issue 3, pp 375-383.
- [16] Djamila Mechta and Saad Harous. Prolonging WSN lifetime using a new scheme for Sink moving based on Artificial Fish Swarm Algorithm. Computer Science Department

Ferhat ABBES University, Setif-1, Algeria. College of Information Technology UAE University, Al Ain, UAE.

[17] Brownlee J. “Clever Algorithms Nature-Inspired Programming Recipes”, Lulu Enterprises, Australia, pp 257-264, 2011.

[18] L. He, J. Xu, Y. Yu and M. Li, 2009, “*Genetic algorithm based length reduction of mobile BS paths in WSNs*”. In: *Computer and Information Science, Eighth IEEE/ACIS International Conference on IEEE*, Shanghai, 1-3 June 2009, pp 797 – 802.

[19] X.-S. Yang, “Firefly algorithms for multimodal optimization”, in: *Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences*, Vol. 5792, pp. 169-178 (2009).

[20] N. V. S. N Sarma, Prof &Gopi, Mahesh. (2014). Energy Efficient Clustering using Jumper Firefly Algorithm in Wireless Sensor Networks. *International Journal of Engineering Trends and Technology*. 10. 10.14445/22315381/IJETT-V10P304.

[21] Tang, Rui & Fong, Simon & Yang, Xin-She & Deb, Suash. (2012). Wolf search algorithm with ephemeral memory. 10.1109/ICDIM.2012.6360147.

[22] DW van der Merwe, AP Engelbrecht, «Data Clustering using Particle Swarm Optimization », Department of Computer Science University of Pretoria.

Résumé

Les réseaux de capteurs sans fil (RCSFs) sont, de nos jours, exploité de plus en plus dans différents domaines d'application. Mais ils souffrent toujours du problème de faible capacité et surtout de consommation d'énergie.

La mobilité de la station de base (BS) est considérée comme l'une des solutions possibles à la résolution de la lacune citée précédemment, ce qui poussent les chercheurs à innover sur de nouveaux schémas qui visent à optimiser le mouvement de la BS.

Dans ce travail, nous avons proposé un schéma bio-inspiré pour le mouvement de la BS fondé sur deux algorithmes bio-inspirés connus pour être performants (Firefly Algorithm et Wolf Search Algorithm). Les performances de notre contribution sont testées via un simulateur que l'on a implémenté sous Java.

Mots clés : RCSFs, Firefly, Wolf Search, sink mobile.

Abstract

Wireless sensor networks (WSNs) are nowadays increasingly exploited in different fields of application. But they still suffer from the problem of low capacity, especially of the energy consumption.

The mobility of the base station (BS) is considered as one of the possible solutions to the resolution of the problem mentioned above, which push the researchers to innovate on new schemes that aim to optimize the movement of the BS.

In this work, we have proposed a bio-inspired scheme for the BS movement based on two bio-inspired algorithms known to be efficient (Firefly Algorithm and Wolf Search Algorithm). The performance of our contribution is tested via a simulator that we implemented in Java.

Key words: WSNs, Firefly, Wolf Search, mobile sink.