Paradigme bio-inspiré dans les réseaux intelligents dynamiques au service de l'internet des objets

Ranida Hamidouche

▶ **To cite this version:**

Ranida Hamidouche. Paradigme bio-inspiré dans les réseaux intelligents dynamiques au service de l'internet des objets. Bio-informatique [q-bio.QM]. Université Paris-Saclay; Université Ferhat Abbas (Sétif, Algérie), 2021. Français. NNT : 2021UPASG066 . tel-03562069

HAL Id: tel-03562069

https://theses.hal.science/tel-03562069

Submitted on 8 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE DE DOCTORAT

NNT : 2021UPASG066

Paradigme bio-inspiré dans les réseaux intelligents dynamiques au service de l'Internet des objets

Bio-inspired paradigm in dynamic smart networks in the service of the Internet of Things

Thèse de doctorat de l'université Paris-Saclay et de l'université Ferhat Abbas Sétif 1

École doctorale n° 580, sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Informatique
Graduate School : Informatique et sciences du numérique, Référent : Université de Versailles-Saint-Quentin-en-Yvelines

Thèse préparée dans l'unité de recherche LI-PaRAD (Université Paris-Saclay, UVSQ), sous la direction de Mourad GUEROUI, Maître de conférences, HDR, la co-direction de Zibouda ALIOUAT, Professeur des universités

Thèse soutenue à Guyancourt, le 10 Décembre 2021, par

Ranida HAMIDOUCHE

Composition du jury

M. Tijani Chahed Professeur des universités, Telecom SudParis	Président
Mme Nabila Labraoui Professeur des universités, Université de Tlemcen	Rapportrice & Examinatrice
M. Hakim Mabed Maître de conférences, Université de Franche-Comté	Rapporteur & Examinateur
M. Makhlouf Aliouat Professeur des universités, Université Ferhat Abbas Sétif 1	Examinateur
M. Ado-Adamou Abba-Ari Maître de conférences, Université Versailles St-Quentin en Yvelines	Examinateur
Mme Sondès Khemiri-Kallel Maître de conférences, Université Versailles St-Quentin en Yvelines	Examinatrice
M. Abdelhak Gueroui Maître de conférences, Université Versailles St-Quentin en Yvelines	Directeur de thèse
Mme Zibouda Aliouat Professeur des universités, Université Ferhat Abbas Sétif 1	Co-Directrice de thèse

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Paradigme bio-inspiré dans les réseaux intelligents dynamiques au service de l'Internet des objets
Mots clés : Internet des objets, réseau de capteurs sans fil mobile, hétérogénéité des capteurs, bio-inspiration

Résumé : Aujourd'hui l'Internet permet de connecter des milliards d'appareils électroniques hétérogènes et assure la communication entre eux. Ces appareils sont dotés de capteurs conçus avec des contraintes en ressources qui affectent considérablement la collecte de données, en particulier une limitation de taille de mémoire et de batterie. Les divergences des caractéristiques de ces objets exigent de nouvelles méthodes intelligentes afin d'assurer la communication entre eux. Les solutions heuristiques deviennent obsolètes ou impuissantes pour satisfaire l'exigence de l'utilisateur, d'où une recherche de nouvelles méthodes devient obligatoirement nécessaires afin de satisfaire les usagers. Parmi ces solutions, nous avons celles basées sur les modèles bio-inspirés. Dans ce contexte, ayant comme objectif la minimisation de la perte de données, nous proposons différentes approches bio-inspirées pour la mobilité du collecteur de données et le choix des points rendez-vous pour la collecte de données. Tout d'abord, nous proposons une mobilité inspirée du mouvement des bactéries Eschericia coli. Ensuite, nous proposons une technique inspirée du regroupement des loups gris et des baleines pour le choix des points rendez-vous. Après, nous améliorons la mobilité du collecteur par une mobilité inspirée des salpes. Enfin, nous combinons ces approches en prenant en considération l'urgence des données. Les différentes contributions ont été évaluées en utilisant des simulations et ensuite ont été comparé avec des travaux similaires existants. Les résultats obtenus sont très prometteurs en termes de consommation d'énergie et de quantité de données collectées.

Title : Bio-inspired paradigm in dynamic smart networks in the service of the Internet of Things
Keywords : Internet of things, mobile wireless sensor network, sensor heterogeneity, bio-inspiration

Abstract : Today the Internet makes it possible to connect billions of heterogeneous electronic devices and ensures communication between them. These devices have sensors designed with resource constraints that significantly affect data collection, especially memory and battery size limitations. The divergences in the characteristics of these objects require new intelligent methods to ensure communication between them. Heuristic solutions become obsolete or powerless to satisfy the user's requirement, hence a search for new methods becomes necessarily necessary to satisfy users. Among these solutions, we have those based on bio-inspired models. In this context, intending to minimize data loss, we propose different bio-inspired approaches for the mobility of the data collector and the choice of meeting points for data collection. First of all, we propose mobility inspired by the movement of Escherichia Coli bacteria. Then, we propose a technique inspired by the grouping of grey wolves and whales for the choice of meeting points. Afterward, we improve the mobility of the collector by mobility inspired by salps. Finally, we combine these approaches by taking into account the urgency of the data. The different contributions were evaluated using simulations and then were compared with similar existing work. The results obtained are very promising in terms of energy consumption and the amount of data collected.

Résumé

Aujourd'hui l'Internet permet de connecter des milliards d'appareils électroniques hétérogènes et assure la communication entre eux. Ces appareils sont dotés de capteurs conçus avec des contraintes en ressources qui affectent considérablement la collecte de données, en particulier une limitation de taille de mémoire et de batterie. Les divergences des caractéristiques de ces objets exigent de nouvelles méthodes intelligentes afin d'assurer la communication entre eux. Les solutions heuristiques deviennent obsolètes ou impuissantes pour satisfaire l'exigence de l'utilisateur, d'où une recherche de nouvelles méthodes devient obligatoirement nécessaires afin de satisfaire les usagers. Parmi ces solutions, nous avons celles basées sur les modèles bio-inspirés. Dans ce contexte, ayant comme objectif la minimisation de la perte de données, nous proposons différentes approches bio-inspirées pour la mobilité du collecteur de données et le choix des points rendez-vous pour la collecte de données. Tout d'abord, nous proposons une mobilité inspirée du mouvement des bactéries Eschericia coli. Ensuite, nous proposons une technique inspirée du regroupement des loups gris et des baleines pour le choix des points rendez-vous. Après, nous améliorons la mobilité du collecteur par une mobilité inspirée des salpes. Enfin, nous combinons ces approches en prenant en considération l'urgence des données. Les différentes contributions ont été évaluées en utilisant des simulations et ensuite ont été comparé avec des travaux similaires existants. Les résultats obtenus sont très prometteurs en termes de consommation d'énergie et de quantité de données collectées.

Mots clés : Internet des objets, réseau de capteurs sans fil mobile, hétérogénéité des capteurs, bio-inspiration

Abstract

Today the Internet makes it possible to connect billions of heterogeneous electronic devices and ensures communication between them. These devices have sensors designed with resource constraints that significantly affect data collection, especially memory and battery size limitations. The divergences in the characteristics of these objects require new intelligent methods to ensure communication between them. Heuristic solutions become obsolete or powerless to satisfy the user's requirement, hence a search for new methods becomes necessarily necessary to satisfy users. Among these solutions, we have those based on bio-inspired models. In this context, intending to minimize data loss, we propose different bio-inspired approaches for the mobility of the data collector and the choice of meeting points for data collection. First of all, we propose mobility inspired by the movement of Escherichia Coli bacteria. Then, we propose a technique inspired by the grouping of grey wolves and whales for the choice of meeting points. Afterward, we improve the mobility of the collector by mobility inspired by salps. Finally, we combine these approaches by taking into account the urgency of the data. The different contributions were evaluated using simulations and then were compared with similar existing work. The results obtained are very promising in terms of energy consumption and the amount of data collected.

Keywords : Internet of things, mobile wireless sensor network, sensor heterogeneity, bio-inspiration

Remerciements

Je tiens à remercier vivement en premier lieu **Dr. Abdelhak Gueroui**, mon directeur de thèse, de m'avoir accueillie dans son équipe, pour sa sympathie, ses conseils, ses orientations scientifiques et ses encouragements qui m'ont permis de mener à bien cette thèse.

J'adresse mes très sincères remerciements à mon codirecteur de thèse **Pr. Zibouda Aliouat**. Ses conseils précieux, ses remarques pertinentes et son suivi régulier ont été d'une grande importance dans l'aboutissement de ce projet, tant au niveau professionnel qu'au niveau personnel.

Je tiens aussi à remercier les professeurs **Mme Nabila Labraoui** et **M. Hakim Mabed** qui se sont intéressés à mon travail et qui ont accepté de l'évaluer en tant que rapporteurs. Leurs conseils et suggestions ont contribué à l'enrichissement de la thèse. Je remercie également les professeurs **M. Tijani Chahed**, **M. Makhlouf Aliouat**, **Mme Sondès Khemiri-Kallel** et **M. Ado-Adamou Abba-Ari** qui ont accepté d'être membres du jury de soutenance de cette thèse.

Je remercie **Dr. Abba Ari** pour les réunions et les discussions de recherche dans le domaine du Réseau de capteurs sans fil (RCSF) et de la Bio-inspiration, tous les membres du laboratoire LI-PaRAD qui m'ont offert un excellent cadre de travail ainsi qu'un séjour extrêmement agréable.

Sur un plan plus personnel, je remercie **ma mère**, **mon père** et mes sœurs **Ines** et **Melissa** pour leur soutien affectif et moral durant ces trois années.

Je remercie également **mon mari** et mon fils **Mohamed Yanis**. Leur encouragement m'a été d'une grande utilité.

Un grand merci à mes très chers amies **Imene**, **Meriem**, **Manel** et **Asma** avec qui j'ai partagé mon quotidien et qui étaient toujours présents à mes côtés.

Sommaire

Introduction générale 1

I État de l'art sur la bio-inspiration dans les réseaux de capteurs sans fil mobiles appliqués à l'Internet des objets 5

1 Réseaux de capteurs sans fil mobiles et l'Internet des objets 6
- **1.1 Internet des objets (IdO)** . . . 6
- **1.2 Défis de l'IdO** . . . 7
- **1.3 Réseaux de capteurs sans fil (RCSFs) mobiles** . . . 9
- **1.4 Objectif de la mobilité dans les RCSFs** . . . 11
- **1.5 Domaines d'applications des RCSFs mobiles** . . . 11
- **1.6 Défis des RCSFs mobiles** . . . 12
- **1.7 Conclusion** . . . 13

2 La topologie et les modèles de mobilité utilisés dans les RCSFs mobiles 14
- **2.1 Approches heuristiques du regroupement des capteurs** . . . 14
- **2.2 Classification des modèles heuristiques de mobilité** . . . 15
  - 2.2.1 Mobilité aléatoire . . . 16
    - 2.2.1.1 DCMDC (Dynamic clustering to efficiently maintain MDC relay networks) . . . 16
  - 2.2.2 Mobilité maîtrisée et géographique . . . 16
    - 2.2.2.1 CDCMP (Clustering duty cycle mobility aware protocol) . . . 17
    - 2.2.2.2 FPM (Forwarding packet model with MS) et SS (Storing packet at source nodes) . . . 17
  - 2.2.3 Mobilité prévisible . . . 17
    - 2.2.3.1 MQRP (Mobile sinks-based QoS-aware data gathering protocol) . . . 17
  - 2.2.4 D'autres formes de mobilité . . . 18
    - 2.2.4.1 REP (Reduced energy path), RDP (Reduced delay path) et DBP (Delay bound path) . . . 18
    - 2.2.4.2 MDP (Markov decision processus) . . . 19
- **2.3 Limites des modèles heuristiques de mobilité** . . . 20
- **2.4 Bilan** . . . 20
- **2.5 Conclusion** . . . 21

3 La bio-inspiration dans les RCSFs mobiles 22
3.1 Approches bio-inspirées du regroupement des capteurs . . . 22
3.1.1 GWO (Grey wolf optimization) . . . 23
3.1.2 Fuzzy-GWO . . . 23
3.1.3 FIGWO (Fitness value based improved grey wolf optimization) . . . 24
3.1.4 WOA-Clustering . . . 24
3.1.5 CHEI (Cluster head election using imperialistic competition algorithm) . . . 24
3.2 Modèles de mobilité bio-inspirés . . . 25
3.2.1 BFOA (Bacterial foraging optimization algorithm) . . . 25
3.2.2 WPA (Path planning of mobile sink based on wolf pack algorithm) . . . 25
3.2.3 AFSA-MS (Artificial fish swarm algorithm for mobile sink) . . . 25
3.2.4 BI-DFAMDT (Backbone integrated discrete firefly algorithm for mobile data transporter) . . . 26
3.2.5 PSO (Particle swarm optimization) . . . 26
3.3 Bilan . . . 26
3.4 Conclusion . . . 27

II Contributions 29

4 Mouvement du collecteur de données inspiré par les bactéries 30
4.1 Bacterial foraging optimization algorithm (BFOA) . . . 30
4.2 Mobilité du Sink basé sur BFOA (SMBFOA) . . . 32
4.2.1 Phase de regroupement et élection des chefs . . . 32
4.2.2 Phase de routage . . . 32
4.2.2.1 Transmissions intra-groupes . . . 32
4.2.2.2 Transmissions inter-groupes . . . 32
4.2.3 Exemple . . . 34
4.2.4 Évaluation des performances de SMBFOA . . . 35
4.3 Conclusion . . . 39

5 Regroupement inspiré par les loups gris et les baleines 40
5.1 Solutions bio-inspirées . . . 40
5.1.1 GWO . . . 41
5.1.2 WOA . . . 41
5.1.3 ICA . . . 42
5.2 Solutions bio-inspirées hybrides pour le débordement du tampon au niveau des nœuds non chefs . . . 42
5.2.1 La sélection des chefs basée sur GWO . . . 42
5.2.1.1 Recherche de proies (Exploration) . . . 42
5.2.1.2 Encerclement de la proie . . . 43
5.2.1.3 Chasse (Optimisation) . . . 44
5.2.1.4 Attaque de proies (Exploitation) . . . 45

5.2.1.5 Condition de terminaison . . . 46
5.2.2 La sélection des chefs basée sur WOA . . . 48
5.2.2.1 Attaque des proies (Exploitation) . . . 48
5.2.3 L'adhésion des membres aux groupes selon ICA . . . 49
5.2.3.1 Génération des empires initiaux . . . 50
5.2.4 Transmission de données . . . 52
5.3 Exemple . . . 52
5.4 Modélisation et vérification . . . 53
5.5 Modèles . . . 55
5.5.1 Modèle énergétique . . . 55
5.5.2 Modèle de réseau . . . 55
5.5.3 Modèle de système et hypothèses . . . 57
5.6 Évaluation des performances de GWO et WOA . . . 57
5.6.1 Variation du nombre d'itérations . . . 58
5.6.2 Complexité temporelle . . . 58
5.6.3 Variation du paramètre *a* . . . 60
5.6.4 Le passage à l'échelle . . . 60
5.6.5 La fonction de coût vs le nombre d'itérations . . . 60
5.6.6 Réseau homogène vs réseau hétérogène . . . 61
5.6.7 Variation de la localisation de la SB . . . 61
5.6.8 Variation de *x(t+1)* . . . 61
5.6.9 GWO vs WOA vs Fuzzy-GWO . . . 62
5.7 Exemple d'application . . . 63
5.8 Conclusion . . . 66

6 Mouvement du collecteur de données inspiré par les salpes 67
6.1 Modified salp swarm algorithm (MSSA) . . . 67
6.2 Évaluation des performances de MSSA . . . 70
6.2.1 Hypothèses . . . 70
6.2.2 Initialisation du réseau . . . 70
6.2.3 Variation du temps de collecte de données . . . 70
6.2.4 Variation du nombre d'itérations . . . 71
6.2.5 Variation du paramètre α . . . 72
6.2.6 MSSA vs SSA vs SMBFOA . . . 74
6.3 Conclusion . . . 75

7 Collecte de données urgentes dans un réseau hétérogène 77
7.1 Solution pour le débordement du tampon au niveau des chefs . . . 77
7.1.1 Phase de regroupement . . . 77
7.1.1.1 Élection des chefs et adhésion aux groupes . . . 79
7.1.2 Phase de routage . . . 79
7.1.2.1 Transmissions intra-groupes . . . 79
7.1.2.2 Transmissions inter-groupes . . . 79

7.1.3 Traitement de l'urgence des données 81
7.2 Évaluation des performances de CGMSSA 83
7.2.1 Mouvement du collecteur de données 83
7.2.2 GWO vs CG vs CGMSSA 84
7.3 Conclusion 86

Conclusion générale et perspectives **87**

Liste des publications personnelles **90**

Bibliographie **92**

Table des figures **100**

Liste des tables **101**

Liste des algorithmes **102**

Liste des abréviations **103**

III Annexes **106**

A Étude statistique sur les approches heuristiques et bio-inspirées dans un contexte IdO **107**

Introduction générale

Motivation

Avec l'émergence de la technologie Internet des objets (IdO), un grand nombre d'applications basées sur des capteurs sont déployées. Ainsi, en raison de leurs implications primordiales, ces capteurs doivent fonctionner de manière cohérente et remplir leurs tâche efficacement. Pour cela, les communications qu'ils impliquent doivent être optimales pour répondre aux exigences attendues des utilisateurs de l'IdO.

La mise en réseau des capteurs reste un problème critique vu leurs ressources limitées. Selon Patel et al. [1], l'énergie consommée par les communications est significative par rapport à celle consommée par les calculs. Elle dépend généralement de la portée de transmission et de l'exposant d'atténuation. Le routage multi-sauts a été introduit pour réduire la distance de transmission et fournir les données détectées en passant par plusieurs nœuds jusqu'à ce qu'elles atteignent la Station de base (SB). Ce type de routage utilise plusieurs relais au lieu de communications à un seul saut, et donc consomme moins d'énergie. Par contre, il introduit un délai plus élevé [2]. De plus, il fait face à ce qu'on appelle le problème de nœuds surchargés (Hotspot problem) [3]. Cela signifie que les nœuds proches de la SB épuisent leur énergie rapidement par rapport aux autres nœuds. Cela est dû à la concentration du trafic de données vers la SB. Par conséquent, pour réduire le nombre de nœuds surchargés, les scientifiques ont introduit l'utilisation des collecteurs de données mobiles [2]. Ces derniers collectent les données des autres capteurs et réduisent l'énergie consommée lors des transmissions. Les données sont ensuite envoyées à la SB via Internet ou par satellite [4].

Les collecteurs de données mobiles permettent d'améliorer considérablement l'économie d'énergie au niveau des nœuds capteurs vu que la collecte de données se fait à proximité des nœuds. Ils offrent aussi un équilibrage de charge et une consommation d'énergie uniforme. Ils réduisent également le nombre de sauts nécessaires pour transmettre les données des capteurs à la SB spécifiquement dans les applications temps réel. Ce qui réduit non seulement les conflits et les collisions, mais également la perte de messages. De plus, ils peuvent couvrir les régions isolées afin que la contrainte de connectivité du réseau soit allégée. La mobilité des collecteurs de données augmente le débit de données et la fiabilité en consommant moins d'énergie à cause de l'utilisation des plus courts chemins. Le problème des nœuds surchargés autour du collecteur est efficacement atténué. En outre, ils peuvent évoluer au sein de plusieurs réseaux. Cette mobilité est l'un des éléments clés de l'IdO.

La conception d'une stratégie qui permet aux collecteurs de données mobiles de se déplacer de manière autonome, distribuée et auto-organisée n'est pas évidente à réaliser par un algorithme polynomial déterministe.

Les solutions biologiquement inspirées du comportement collectif des communautés sociales fournissent des outils alternatifs et des algorithmes efficaces qui émergent de nombreuses propriétés intéressantes applicables à la technologie des capteurs. Ces solutions implémentent des systèmes hautement efficaces, structurellement simples, puissants, distribués et tolérants aux pannes. Certaines sociétés biologiques, comme les colonies de la bactérie Escherichia coli [1], ou les salpes [2], offrent des perspectives à certains capteurs mobiles pour acquérir une intelligence artificielle leur permettant de se déplacer de façon autonome à travers le réseau.

Intuitivement, nous pouvons laisser le collecteur de données visiter chaque objet, et le problème sera le même que le problème NP-complet du voyageur de commerce (Traveling salesmen problem (TSP)). Cependant, lorsque l'échelle du réseau est grande, une telle solution peut être irréalisable car le collecteur de données est susceptible de violer la contrainte de délai de collecte de données en raison du déplacement sur un chemin beaucoup plus long. Alternativement, on peut sélectionner un sous-ensemble d'objets comme Points rendez-vous (PRs) pour réduire la longueur du chemin. Les objets non PRs envoient des données aux PRs voisins et le collecteur de données se déplace pour visiter uniquement les PRs afin de collecter toutes les données. Ainsi, le problème devient alors comment trouver ces PRs de telle sorte que la consommation d'énergie due aux communications multi-sauts (des objets non PRs aux PRs) soit minimisée en parcourant un chemin de déplacement plus court. Ce type de problème n'a pas pu être résolu en temps polynomial en utilisant des méthodes conventionnelles. Les algorithmes bio-inspirés pourraient être à nouveau une alternative efficace capable de trouver des solutions de bonnes qualités en un temps raisonnable [5].

Cette thèse s'intéresse essentiellement aux approches bio-inspirées visant à optimiser la collecte de données dans un Réseau de capteurs sans fil (RCSF) hétérogène en utilisant un collecteur de données mobile. Nous visons les applications de surveillances dans le domaine de E-santé, ou les données sont collectées de manière périodique ou événementielle. Dans ce type d'application, en cas d'urgence, la perte de données et le temps d'acheminement de ces données sont intolérables afin d'éviter les catastrophes. Nous abordons le problème de perte de données du au débordement des tampons des capteurs hétérogènes. Plus particulièrement, nous nous intéressons à deux aspects : le mouvement du collecteur de données qui joue un rôle très important dans la collecte de données ; et le choix de l'emplacement des PRs qui permet d'améliorer la trajectoire du collecteur de données, et ainsi les performances du réseau.

Contributions et Organisation

Le but de notre travail est d'aborder le problème de la collecte de données dans les réseaux IdO supportant l'hétérogénéité, les ressources limitées des objets et le passage à l'échelle du réseau. Nous nous sommes focalisés sur le modèle de mobilité utilisé par le collecteur de données ainsi que la technique choisie pour sélectionner les PR en visant à minimiser la perte de données et la consommation énergétique.

Pour arriver à ce but, nous avons commencé par proposer une mobilité imitant le mouvement des bactéries Escherichia coli, appelée SMBFOA. Cette idée était testée dans une topologie hiérarchique conventionnelle

1. Escherichia coli est une bactérie intestinale des mammifères, très commune chez l'être humain.
2. Les salpes, ou salpidae (sea salp en anglais), sont des animaux marins qui se déplacent par contractions, pompant l'eau via leur corps gélatineux.

dans un réseau homogène comportant des capteurs ayant la même taille du tampon et même énergie initiale. Le choix d'utiliser ce type de topologie est dû au fait que la division du réseau en sous-zones (groupes) augmente la durée de vie des capteurs. De plus, cela permettra d'avoir moins de nœuds à visiter par le collecteur de données mobile. Ce qui réduit le temps passé avec les nœuds et augmente par conséquent la fréquence de ses visites aux PRs. De plus, la durée d'une tournée est considérablement réduite lors de l'utilisation des PRs. Les résultats étaient très prometteurs lorsque la mobilité intelligente a été introduite. Sauf que la solution ne prenait pas en considération le problème du débordement du tampon des capteurs.

Ensuite, nous avons amélioré l'approche par un regroupement inspiré de la vie commune des loups gris (GWO) et des baleines (WOA). Cette méthode prend en compte la taille restreinte de la mémoire des capteurs et elle est applicable dans un contexte IdO (réseau hétérogène). Les PRs ont des tampons suffisamment grands pour mettre en cache leurs données détectées et les données reçues par leurs voisins avant que le collecteur de données ne vienne les chercher. Ce qui réduit considérablement la perte de données.

Par la suite, nous avons perfectionné la mobilité du collecteur de données par une stratégie récente inspirée des salpes, réduisant ainsi la complexité de l'algorithme.

Enfin, en testant cette solution, nous avons remarqué que le temps de la collecte de données a été minimisé mais pas au point d'être adapté à une application critique. Donc, nous avons élaboré une solution qui peut être applicable dans une situation critique en temps réel qui prend en considération l'urgence des données.

La thèse est organisée comme suit :

- Le premier chapitre introduit des généralités sur les Réseaux de capteurs sans fil (RCSFs) mobiles appliqués à l'IdO tout en précisant l'intérêt de l'utilisation de cette mobilité. Il présente leurs domaines d'applications et les défis rencontrés au sein de ces réseaux.
- Le deuxième chapitre relate les approches heuristiques du regroupement des capteurs, des classifications des modèles de mobilités heuristiques ainsi que les limites de ces modèles.
- Le troisième chapitre présente la bio-inspiration dans les RCSFs mobiles pour le regroupement des capteurs et la mobilité du collecteur de données sur lesquels notre travail est basé. Les approches existantes concernant ces deux aspects sont résumées et analysées. Un bilan soulignant les limites de chaque approche étudiée est dressé.
- Dans le quatrième chapitre, nous présentons la première contribution qui consiste en un algorithme inspiré du mouvement des bactéries Escherichia coli pour le déplacement du collecteur de données dans un réseau homogène.
- Dans le cinquième chapitre est présentée la technique pour le choix des PRs inspiré de la vie hiérarchique des loups gris et des baleines en prenant en considération le problème du débordement du tampon des nœuds non PRs. Après la phase de regroupement, les nœuds envoient leurs données au chef concerné en utilisant un routage multi-saut intra-clusters. Le groupe, dans ce cas, est subdivisé en régions et zones. Chaque membre sait, qui sont ses prédécesseurs pour attendre de recevoir leurs données, et qui sont ses successeurs pour transmettre leurs données au niveau supérieur. Une fois que le chef a reçu toutes les données de ses membres, il les envoie à la SB.

- Dans le sixième chapitre nous présentons un nouveau modèle de mobilité du collecteur de données inspiré du mouvement des salpes dans un contexte IdO.
- Le septième chapitre présente une approche de collecte de données urgentes en prenant en considération le débordement du tampon des PRs.
- Nous terminons avec une conclusion générale résumant nos contributions présentées dans cette thèse et nous introduisons nos perspectives pour les prochains travaux.

Première partie

État de l’art sur la bio-inspiration dans les réseaux de capteurs sans fil mobiles appliqués à l’Internet des objets

Chapitre 1

Réseaux de capteurs sans fil mobiles et l'Internet des objets

Sommaire

1.1 Internet des objets (IdO) 6
1.2 Défis de l'IdO 7
1.3 Réseaux de capteurs sans fil (RCSFs) mobiles 9
1.4 Objectif de la mobilité dans les RCSFs 11
1.5 Domaines d'applications des RCSFs mobiles 11
1.6 Défis des RCSFs mobiles 12
1.7 Conclusion 13

Actuellement, nous vivons le début d'une nouvelle révolution industrielle. La croissance des machines et des objets connectés appelés IdO rivalisera avec les merveilles technologiques du passé. Le RCSF est l'une des composantes fondamentales du développement de l'IdO pour les Technologies de l'information et de la communication (TIC). Pour faire face aux problèmes rencontrés lors de l'intégration des capteurs à l'IdO, les chercheurs ne se limitent plus aux solutions heuristiques, mais ouvrent de nouvelles perspectives prometteuses à l'utilisation de ce que nous appelons les "inspirations biologiques".

Dans ce chapitre, nous présentons les bases théoriques et les éléments bibliographiques essentiels pour la suite de notre travail. Nous commençons par évoquer quelques généralités concernant l'IdO et les RCSFs mobiles, tout en soulignant les problèmes qui motivent la recherche dans ces domaines.

1.1 Internet des objets (IdO)

De nos jours, la population mondiale augmente dans le milieu urbain et les besoins en termes de services de TIC, ainsi que d'infrastructures de réseau, évoluent rapidement [6]. Pour répondre à ces besoins, des milliers d'appareils intelligents indépendants, notamment des smartphones, des capteurs, des actionneurs, des Radio frequency identification (RFID), des appareils de mesure et des ordinateurs, qui font partie de

notre vie quotidienne, sont mis en place pour fournir une interaction intelligente de manière auto-orchestrée et auto-organisée, pour former le futur Internet. De plus, tous ces appareils hétérogènes sont capables de générer et de fournir des données, dans lesquelles un certain nombre de techniques et de schémas sont impliqués dans la collecte, la transmission et le stockage de données[7, 8]. Ce type d'interaction d'objets intelligents est appelé l'IdO.

L'évolution de la micro-électronique au cours des dernières décennies a permis le développement de capteurs à faible coût et à faible puissance, qui peuvent être déployés sur un environnement à grande échelle pour collecter collectivement les informations du champ déployé et livrer sans fil les données recueillies à un ordinateur distant pour le traitement ou le stockage [9, 10, 11, 12]. Ce cycle est résumé dans la figure 1.1.

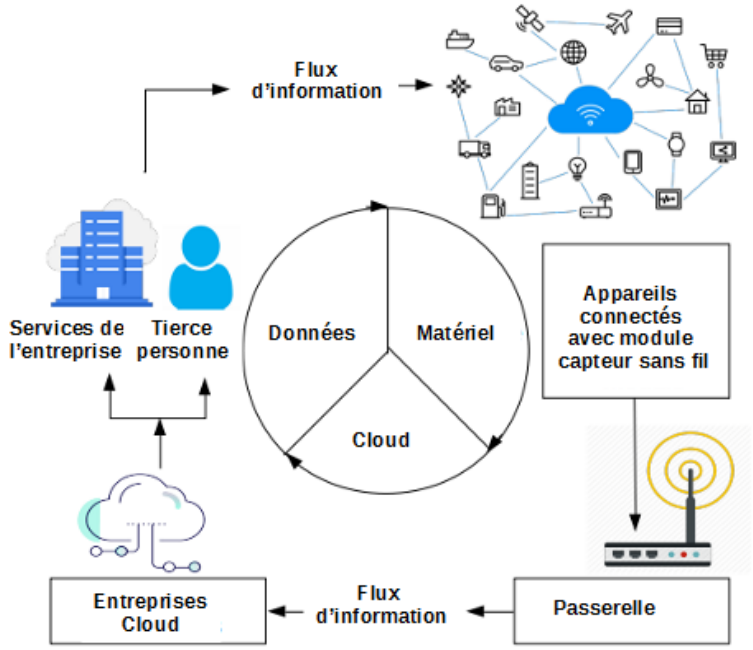


Figure 1.1 – Structure du marché mondial de l'IdO

1.2 Défis de l'IdO

Dans ce qui suit, nous examinons brièvement les défis les plus connus de l'IdO.

- **Énergie**

  La croissance future de l'IdO est menacée par les besoins énergétiques des applications embarquées. Les architectures réseaux de nouvelle génération sont confrontées à des contraintes de ressources et à des limitations structurelles et physiques. Les réseaux sont composés de nœuds contraints en termes d'énergie et de ressources de communication comme dans les RCSFs mobiles [13]. Les systèmes biologiques existants fournissent des informations précieuses aux chercheurs pour trouver un compromis entre les ressources limitées des objets mobiles et la forte demande de l'utilisateur. Par

exemple, dans le processus de recherche de nourriture, les fourmis utilisent leurs ressources limitées pour optimiser le comportement global des colonies afin de trouver une source de nourriture d'une manière rentable.

- **Défis de la technologie et des normes**
  Un environnement collaboratif combine différents appareils, écosystèmes et technologies. Cela rend très difficile l'adoption de toute spécification ou l'aboutissement à un accord global. Dans ce cas, il est nécessaire de prendre en considération l'hétérogénéité et l'interopérabilité lors de la conception d'un système IdO [14, 15, 16, 17, 18, 19]. Différents niveaux d'hétérogénéité sont également observés dans les systèmes biologiques. Malgré la diversité des moyens de communication que l'on trouve dans les sociétés d'insectes, les différences de taille ou de travail, ils peuvent communiquer et collaborer facilement.
- **Gestion de données**
  L'interaction de nombreux objets mobiles entraîne une congestion du réseau [16, 20]. De nouvelles technologies doivent être envisagées pour gérer la croissance exponentielle des données fournies par les objets mobiles connectés à Internet. L'utilisation des phénomènes biologiques peut à nouveau être une bonne initiative pour une solution efficace. Les communications sont toujours maintenues pour les animaux vivants en grands groupes.
- **Flexibilité**
  Les objets doivent être flexibles pour être adaptés à différents contextes [15]. Dans la nature, les individus de la même espèce peuvent rencontrer des conditions environnementales très variables. Ainsi, un animal vivant dans une zone urbaine, agricole ou forestière n'aura pas accès aux mêmes ressources et ne sera pas exposé aux mêmes polluants. Mais, il s'adaptera à l'environnement dans lequel il vit. On peut s'inspirer de ce phénomène qui a prouvé son efficacité dans la nature, pour proposer des systèmes IdO adaptatifs aux changements.
- **Disponibilité et Fiabilité**
  La disponibilité fait référence à la capacité de l'IdO à fournir des services à tout le monde à différents endroits simultanément [16]. La fiabilité fait référence au bon fonctionnement du système basé sur sa spécification.
- **Problème de passerelle**
  La passerelle établit un lien entre les objets IdO, les capteurs, le système et le Cloud afin de faciliter leur communication [21]. Une défaillance dans la passerelle perturbera les échanges entre ces composants.
- **Sécurité et confidentialité**
  La vie privée est plus sérieusement menacée en raison de la traçabilité des objets via l'IdO [17, 14]. Le système immunitaire artificiel (Artificial immune system (AIS)) est une bonne base pour développer des solutions efficaces pour faire face aux problèmes de sécurité et de confidentialité.
- **Passage à l'échelle**
  Le passage à l'échelle de l'IdO fait référence à la possibilité d'ajouter de nouveaux appareils, services et fonctions pour les clients sans affecter négativement la qualité des services existants. La structure IdO doit avoir une capacité d'extension des ressources et du nombre d'utilisateurs. Sa taille peut être étendue si nécessaire en raison du nombre potentiellement illimité d'objets [15, 16, 22]. La croissance

et la production bactériennes se multiplient dans un environnement adéquat et riche sans affecter négativement la population où elles vivent.

- **Mobilité**
  Connecter les utilisateurs avec leurs services souhaités en continu lors des déplacements est une prémisse importante de l'IdO [16]. Dans le monde de la biologie, chaque être vivant a sa façon de se déplacer pour répondre à ses besoins quotidiens en termes de nourriture. Leur déplacement est toujours bénéfique en énergie et en temps. Même en mouvement, les communications avec le monde qui les entoure ne sont jamais interrompues.

Le but de notre étude est d'assurer l'acheminement des données de n'importe quel capteur vers la SB sans perte de données. C'est pourquoi nous nous intéressons principalement à la mobilité du collecteur de données qui affecte considérablement ce but.

1.3 Réseaux de capteurs sans fil (RCSFs) mobiles

Les RCSFs consistent en un petit ou grand ensemble de dispositifs distincts, tels que les RFID, les capteurs et les actionneurs, etc., qui collectent des informations de la région de déploiement, traitent les données collectées et les communiquent aux utilisateurs finaux [23, 24].

Nous distinguons trois formes de mobilité dans les RCSFs : la mobilité des capteurs, la mobilité des événements et la mobilité du collecteur de données [25, 26, 27]. La mobilité peut être intégrée différemment : les nœuds ont un module de mobilité intégré (moteur ...) ; ils sont embarqués sur un élément qui bouge (être humain, navette de transport ...) ; ou ils se déplacent par une force extérieure (vent, inondation ...).

1. **Mobilité des capteurs**
   La mobilité des capteurs est limitée à quelques applications, telles que les applications militaires (capteurs attachés aux soldats), les applications de surveillance des animaux d'élevage, les oiseaux migrateurs (les capteurs sont portés par les animaux) ou la surveillance des milieux marins (les capteurs suivent le mouvement du courant).
2. **Mobilité des évènements**
   La mobilité des évènements existe principalement dans les applications de suivi de cibles. Dans ce type d'application, il est important que l'événement observé doit être couvert par un nombre suffisant de nœuds. En conséquence, les nœuds se réveilleront autour de l'objet, le surveilleront avec une activité élevée et le capteur passera en mode veille lorsque la cible s'éloignera. Alors que la source d'événements se déplace dans le réseau, elle s'accompagne d'un secteur d'activité dans le réseau qui la suit.
3. **Mobilité du collecteur de données**
   Le collecteur de données est un nœud particulier qui collecte les données des différents nœuds du réseau, doté d'une énergie illimitée lui permettant d'être toujours actif à l'arrivée aléatoire des informations [4]. Il existe principalement trois types de collecteurs de données :

   a- Un nœud appartenant au réseau comme n'importe quel autre nœud (Figure 1.2 partie a).

b- Une entité en dehors du réseau. Par exemple, un ordinateur portable ou un assistant personnel numérique (Personal digital assistant (PDA)) interagissant avec le réseau (Figure 1.2 partie b).

c- Une passerelle vers un autre réseau (Internet par exemple), où la demande d'informations provient d'un centre de traitement distant (Figure 1.2 partie c).

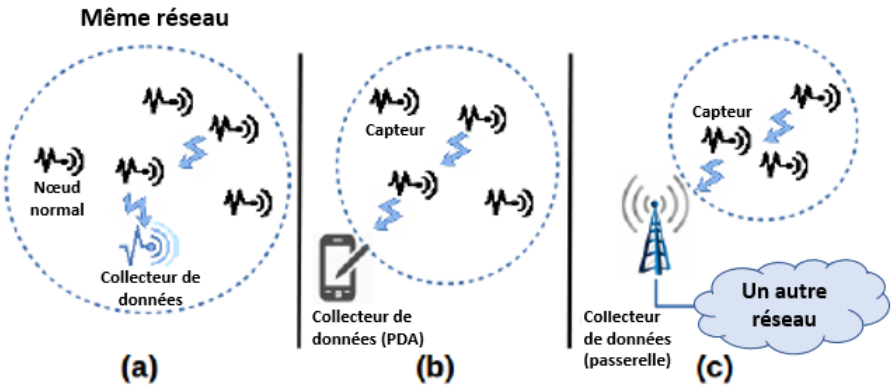


Figure 1.2 – Les types de collecteurs de données

La mobilité du collecteur de données est un cas particulier de mobilité, où seul le point de collecte est mobile. Les capteurs peuvent attendre son arrivée pour lui envoyer leurs données directement ou en utilisant un nombre réduit de sauts. Cela réduira considérablement le trafic et la consommation d'énergie sur le réseau.

La figure 1.3 présente un graphe qui modélise un réseau comportant deux nœuds (N_1, N_2) et un collecteur de données éventuellement situé en L_1 ou L_2. Dans cet exemple, la distance entre les deux nœuds est de $8m$. Chaque nœud a 100 unités d'énergie principalement, il envoie 1 bit/s et consomme 1 unité/m. Dans le premier cas le collecteur de données est statique et positionné en O, au milieu de (N_1, N_2), distant de $4m$ de chaque nœud. Les deux nœuds consomment 4 unités d'énergie pour envoyer 1 bit de données. Puisque chaque nœud a 100 unités d'énergie, il envoie 25 bits avant de s'éteindre ($100/4$ unités). Il est évident que la durée de vie optimale est de 25 secondes. Dans le second cas, le collecteur de données est mobile. Il passe de L_1 à L_2 ou l'inverse. Par exemple, il reste sur L_1 le temps suffisant pour que le nœud N_1 consomme $1/2$ de son énergie (de même pour L_2). Selon la position du collecteur de données mobile, que ce soit à l'emplacement L_1 ou L_2, les nœuds consomment 1 ou 7 unités d'énergie pour transférer 1 bit de données. Le nœud N_1 envoie 50 bits ($50/1$ unités) si le collecteur de données est en L_1 et 7 bits ($50/7$ unités) si le collecteur de données est en L_2. Idem pour le nœud N_2, il envoie 50 bits ($50/1$ unités) si le collecteur de données est en L_2, et 7 bits ($50/7$ unités) si le collecteur de données est en L_1. Chaque nœud envoie 57 bits avant de s'éteindre. Les nœuds n'envoient pas de données lorsque le collecteur de données est en mouvement pour changer sa position. Ainsi, la durée de vie moyenne est de 57 secondes, une augmentation significative par rapport au premier cas, lorsque le collecteur de données est statique.

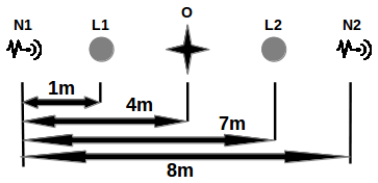


Figure 1.3 – Un graphe modélisant un réseau comportant 2 capteurs et un collecteur de données

1.4 Objectif de la mobilité dans les RCSFs

Il existe quatre principaux objectifs de la mobilité du collecteur de données : la collecte de données, la couverture, la connectivité et l'alimentation des nœuds en énergie.

1. **Collecte de données :** Les collecteurs ont pour rôle de transporter les informations provenant des nœuds sources et éventuellement de les agréger. Ils se déplacent dans la zone d'intérêt et collectent les données en utilisant une communication à courte portée consommant moins d'énergie. Dans ce cas, ils disposent d'une puissance de calcul plus élevée ainsi que d'une mémoire de grande capacité pour traiter directement les données reçues [4, 28].
2. **Couverture :** La qualité de service (Quality of service (QoS)) d'un RCSF peut être mesurée par le taux de couverture. Dans ce contexte, les collecteurs de données mobiles ont pour but de couvrir les zones isolées causées par la mauvaise répartition des capteurs. Ils communiquent avec les nœuds qui n'ont pas de voisins pour ne pas être déconnectés du réseau [4].
3. **Connectivité :** Les RCSFs sont généralement constitués d'un nombre massif de capteurs assurant une bonne connectivité du réseau. Néanmoins, compte tenu des fortes contraintes énergétiques des capteurs, les nœuds peuvent disparaître du réseau en le partitionnant en plusieurs composants associés. Ainsi, les liens de bout en bout, du collecteur de données à la source, disparaissent. Dans ce cas, l'utilisation d'un collecteur de données mobile améliore la connectivité du réseau en permettant la collecte de données à partir de plusieurs composants isolés du réseau [29, 30, 31, 32].
4. **Chargement :** Tous les schémas de rechargement existants reposent sur l'utilisation d'un chargeur mobile qui traverse tous les points du réseau et les recharge. Le chargement sans fil est modélisé par un problème d'optimisation de chemin pour le chargeur mobile, avec un objectif qui minimise le nombre de ces arrêts [4].

1.5 Domaines d'applications des RCSFs mobiles

Le RCSF mobile est appliqué dans la vie courante dans différents domaines. Parmi lesquelles nous citons :

1. **Les applications militaires**
   Le laboratoire national Lawrence Livermore [4] a développé un projet pour la détection de dispositifs nucléaires appelé le système de suivi à grande surface (Wide area tracking system (WATS)). Dans ce système, les capteurs réduisent à la fois les fausses alarmes et améliorent le taux de détection, ce qui facilite le déploiement.

2. **Les applications médicales**
   L'application "Blue Code" [33] est un exemple de projet proposée pour la surveillance médicale. Un autre exemple à citer est la nouvelle plate-forme de surveillance des patients cardiaques utilisant des capteurs pour la collecte de données à partir d'un électrocardiogramme (Electrocardiogram (ECG)) et d'un téléphone portable pour la détection de pathologies cardiaques [4].
3. **Les applications environnementales**
   Une implémentation d'un RCSF mobile sur le volcan "Reventator" dans la partie ouest de l'Amazonie en Équateur a été proposée. Un autre exemple d'applications environnementales est le projet ARGO. Le but de ce projet est de surveiller l'eau des océans. Les nœuds utilisés sont équipés de capteurs de différentes mesures déployés depuis des navires. Les données collectées lors de leurs déplacements sont envoyées à un satellite [4].
4. **Les applications dans le sport**
   Le RCSF mobile est de plus en plus utilisée dans le domaine du sport, à savoir les systèmes de surveillance, les systèmes de calcul de trajectoires (par exemple, dans le tennis, ils utilisent un système appelé "Hawk-Eye" : l'œil du faucon), les systèmes de détection d'erreurs d'arbitrage (dans le football, ils indiquent si le ballon a franchi la ligne de but) [4].
5. **Surveillance écologique de l'habitat**
   Parmi les applications proposées pour la surveillance écologique de l'habitat, le système Zebranet. Il s'agit d'un projet dirigé par l'Université de Princeton et déployé dans le centre du Kenya. Il vise à étudier les populations de zèbres à l'aide de capteurs fixés sur les animaux [34].

1.6 Défis des RCSFs mobiles

Le RCSF mobile est confronté à une variété de défis, tels que le déploiement, la gestion de la mobilité, l'emplacement de l'élément mobile, la navigation mobile, la minimisation de la consommation d'énergie lors du changement de position, le maintien de la connectivité du réseau ainsi que la collecte de données [4].

- **Le déploiement**
  Dans les environnements hostiles, les capteurs sont déployés au hasard et en grand nombre. Ce qui engendre une distribution non uniforme sur le champ de détection, et déclenche par la suite de nouveaux problèmes dans le RCSF tels que la localisation, la connectivité, la couverture, la sécurité, le coût et la durée de vie [35, 36, 37].
- **La qualité de service (QoS)**
  Comme le RCSF est utilisé dans des applications critiques en temps réel, il doit fournir un minimum de QoS. Cependant, cela est difficile à atteindre car la topologie du réseau est dynamique en raison de la mobilité des nœuds [28, 35, 38].
- **La localisation**
  Dans un RCSF mobile, les nœuds n'ont aucune connaissance préalable de la position actuelle du collecteur de données. Ils doivent obtenir sa position pour pouvoir communiquer avec lui [4, 38].
- **La consommation d'énergie**
  La communication sans fil induit un coût énergétique élevé et doit être utilisée efficacement. Une

partie de l'énergie du capteur est perdue lors du traitement et de la collecte de données. En raison des conditions géographiques, il devient parfois très difficile de recharger ou de changer les batteries du capteur [35, 39, 38].

- **La collecte et transmission des données**
  La communication n'étant possible que lorsque les nœuds sont en phase de transmission, il est nécessaire de détecter la présence du collecteur de données mobile de manière correcte et efficace [4, 35].
- **La gestion de la mobilité**
  Le trajet, la vitesse et le temps de séjour des collecteurs de données mobiles doivent être définis pour optimiser les performances du réseau [28].

1.7 Conclusion

Les multiples avantages de l'utilisation de la mobilité dans les RCSFs ont rendu les capteurs puissants pour être intégrés dans de grands réseaux. En conséquence, la mobilité devient un facteur supplémentaire qui peut être exploité efficacement pour la conception de protocoles garantissant une meilleure collecte de données.

Le modèle de mobilité utilisé pour assurer la collecte de données au sein des RCSFs mobiles présente quelques limites de performance qui peuvent affecter la collecte de données. De plus, la topologie utilisée dans le réseau a un effet majeur sur les performances du collecteur de données.

Le prochain chapitre présente les différentes formes de répartitions des nœuds dans le RCSF mobile ainsi que la classification des modèles heuristiques de mobilité. La suite du document portera sur les solutions existantes ou que nous proposons afin de minimiser la perte de données dans ces réseaux.

Chapitre 2

La topologie et les modèles de mobilité utilisés dans les RCSFs mobiles

Sommaire

2.1 Approches heuristiques du regroupement des capteurs 14
2.2 Classification des modèles heuristiques de mobilité 15
2.3 Limites des modèles heuristiques de mobilité 20
2.4 Bilan 20
2.5 Conclusion 21

L'utilisation des collecteurs de données mobiles peut alléger la contrainte de haute densité et assurer une connectivité suffisante du RCSF déployé sur une large zone. Ainsi, une architecture réseau hiérarchique devient une option envisageable, évitant ainsi les problèmes de collision et de perte de messages induits par l'utilisation d'un réseau dense.

Dans ce chapitre, nous présentons les différentes techniques heuristiques de regroupement des nœuds capteurs, et nous classifions les modèles heuristiques de mobilité adoptés par le collecteur de données tout en soulignant les limites de chaque modèle.

2.1 Approches heuristiques du regroupement des capteurs

L'une des méthodes pour améliorer les capacités des capteurs consiste à utiliser des schémas de gestion de topologie [40]. Ces schémas se composent de trois étapes : la découverte de topologie, la gestion du cycle de sommeil du capteur et le regroupement des nœuds.

Le regroupement permet de passer d'une topologie plate à une topologie hiérarchique (Figure 2.1). Le regroupement peut être centralisé ou distribué. Les deux méthodes conviennent aux petits réseaux [41]. Cependant, les approches centralisées sont déconseillées dans un réseau à grande échelle. Comme, c'est la SB qui prend la décision du choix des groupes, le nombre de transferts de données entre elle et les capteurs augmentent proportionnellement que la taille du réseau, ce qui épuise l'énergie des capteurs plus rapidement.

Contrairement aux méthodes centralisées, les approches distribuées sont considérées comme plus efficaces vu que c'est les nœuds eux-mêmes qui prennent la décision de la manière de se regrouper [42].

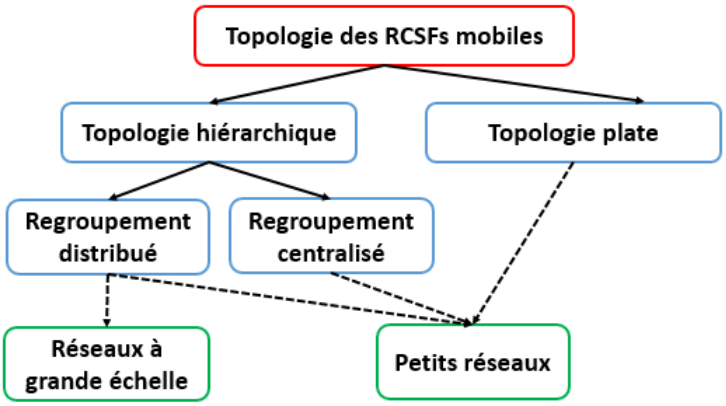


Figure 2.1 – Topologie des RCSFs mobiles

2.2 Classification des modèles heuristiques de mobilité

Un collecteur de données mobile se déplace dans la zone d'intérêt en suivant différentes formes de mobilité. Ces formes de mobilité peuvent être classées en deux familles, les modèles homogènes et les modèles hétérogènes. La première famille est basée sur le mouvement d'un groupe de nœuds ordinaires mobiles suivant le même modèle de mobilité. Cependant, dans la deuxième famille, chaque nœud mobile se déplace selon un modèle de mobilité spécifique. Notant que ces deux familles peuvent être subdivisées à leurs tour comme le montre la figure 2.2.

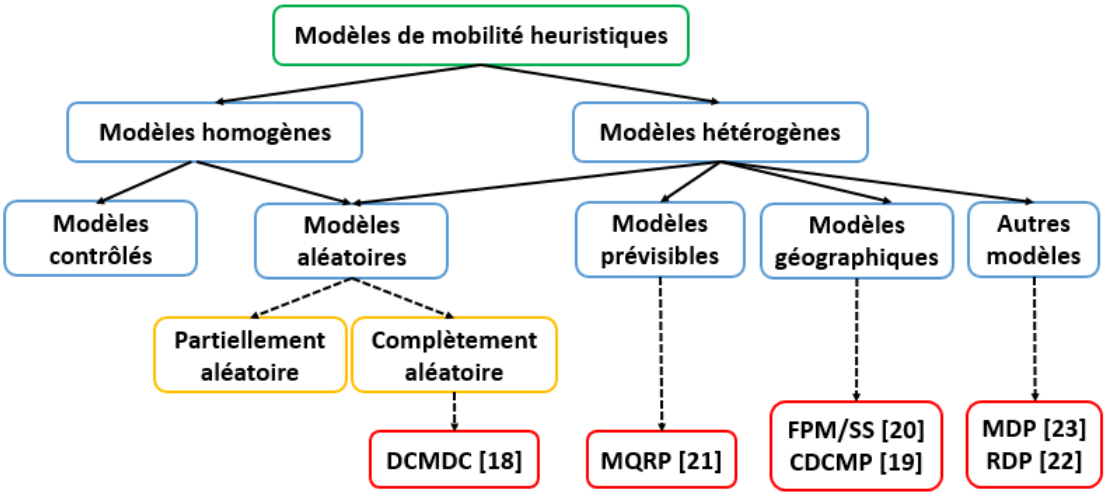


Figure 2.2 – Classification des modèles heuristiques de mobilité

Les modèles de mobilité homogènes peuvent être utilisés lorsqu'il est nécessaire de changer la topologie du réseau pour obtenir les meilleures couverture et connectivité de l'environnement, tandis que les modèles de mobilité hétérogènes peuvent être utilisés lorsqu'il est nécessaire d'avoir un ou plusieurs collecteurs de données mobiles qui se déplacent dans le réseau pour collecter les informations des nœuds statiques.

2.2.1 Mobilité aléatoire

Les modèles de mobilité aléatoires peuvent être utilisés lorsqu'il est nécessaire de simuler le comportement des humains, des animaux et des véhicules [43, 44]. Cette sous-classe est beaucoup exploitée, DCMDC [45] est un exemple de ce type de mobilité.

2.2.1.1 DCMDC (Dynamic clustering to efficiently maintain MDC relay networks)

Abuarqoub et al. [45] ont proposé un regroupement dynamique pour maintenir efficacement les réseaux de relais et superviser les changements topologiques. Ils ont partitionné le réseau en groupes appelés Zones de service (ZSs) pour réduire les paquets de contrôle de mobilité. Ensuite, ils ont placé un collecteur de données dans chaque ZS (Figure 2.3). Le maintien d'une connectivité continue augmente la disponibilité du système. Cependant, les auteurs doivent minimiser le temps passé à se déconnecter de la ZS actuelle et à se connecter à une autre.

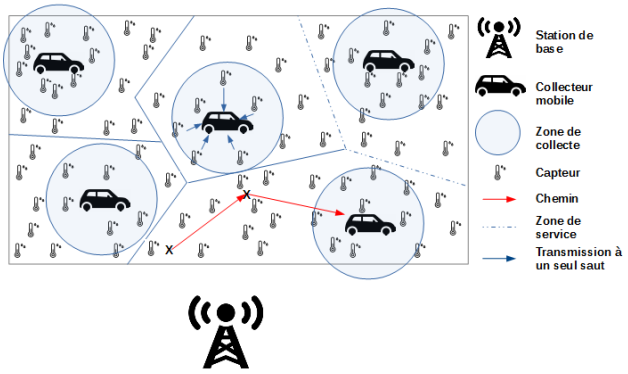


Figure 2.3 – Le protocole DCMDC

2.2.2 Mobilité maîtrisée et géographique

Si le mouvement du collecteur de données est guidé ou contrôlé par un paramètre comme une fonction objective ou par un fait perceptible prédéfini, on dit que la mobilité est géographique ou contrôlée. Cette forme de mobilité peut être utilisée dans le cas où les nœuds mobiles doivent suivre certaines propriétés imposées par l'environnement, comme la présence d'obstacles, lorsque leur mouvement est limité à la zone de déploiement [43, 46]. Ci-dessous, nous présentons deux approches basées sur la mobilité maitrisée et géographique trouvées dans la littérature : CDCMP [47] utilise une topologie héararchique distribuée et FPM/SS [48] utilise une topologie plate.

2.2.2.1 CDCMP (Clustering duty cycle mobility aware protocol)

CDCMP [47] est l'une des solutions proposées pour le mouvement maîtrisé du collecteur de données. Il s'agit d'un protocole de routage hiérarchique, combinant trois aspects très importants dans les RCSFs : un mécanisme duty-cycle avec un collecteur de données mobile dans le contexte d'une organisation en groupes de nœuds. Cependant, l'algorithme utilise une mobilité maitrisée géographique. Le collecteur de données dessine un chemin rectangulaire comme le montre la figure 2.4. La solution hérite de toutes les limites de ce type de mobilité. Le collecteur de données se déplace sur un chemin fixe et ses coordonnées sont aléatoires. Après un certain temps, il peut être éloigné de certains chefs qui souhaitent envoyer des données et privilégier certains d'autres.

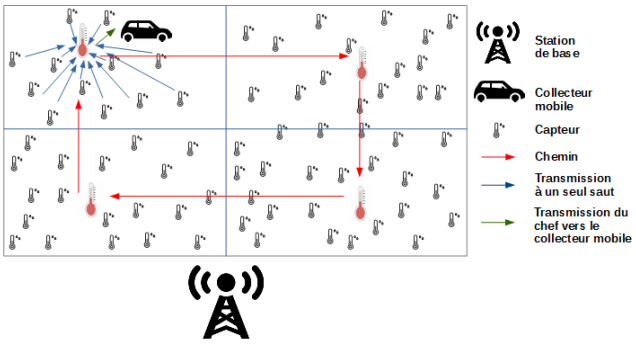


Figure 2.4 – Le protocole CDCMP

2.2.2.2 FPM (Forwarding packet model with MS) et SS (Storing packet at source nodes)

Shakir et al. [48] ont proposé deux systèmes de mise en mémoire tampon pour atténuer le débordement des paquets aux niveau des nœuds et réduire l'encombrement inutile du réseau, le modèle de transfert des paquets nommé Forwarding packet model with MS (FPM) et le stockage de paquets au niveau des nœuds sources nommé Storing packet at source nodes (SS). Ils ont utilisé une communication multi-sauts pour remédier à l'augmentation du retard au niveau des nœuds et réduire la perte de paquets. Cependant, l'utilisation d'une topologie plate rend certaines zones isolées (Figure 2.5).

2.2.3 Mobilité prévisible

Le collecteur de données devine son prochain emplacement en fonction de son historique et communique cette information à son voisinage [43, 44]. Cette sous-classe est peu exploitée et le seul protocole trouvé en faisant partie, MQRP [49], est présenté ci-dessous.

2.2.3.1 MQRP (Mobile sinks-based QoS-aware data gathering protocol)

Faheem et Gungor [49] ont proposé un nouveau protocole de collecte de données pour les applications basées sur les grilles intelligentes (Smart grid (SG)) nommé Mobile sinks-based QoS-aware data gathering protocol (MQRP). Ils utilisent plusieurs collecteurs de données suivant un modèle de mobilité hybride,

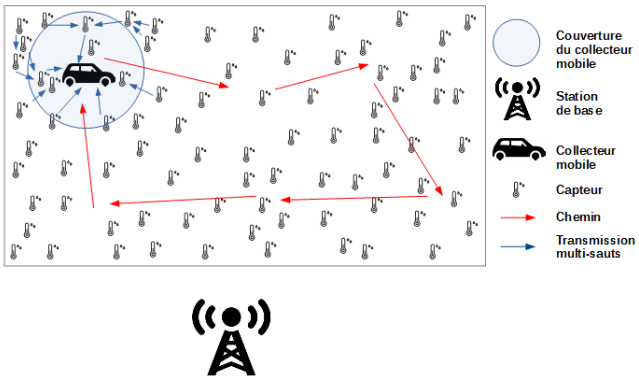


Figure 2.5 – Le protocole FPM/SS

combinant la mobilité prévisible et la mobilité maîtrisée (Figure 2.6). Dans une première étape, la vitesse du collecteur de données est fixe et la trajectoire à suivre est prévisible pour collecter des données à partir d'un sous-réseau. Ensuite, il donne la priorité aux régions actives où les événements se produisent avec une probabilité plus élevée.

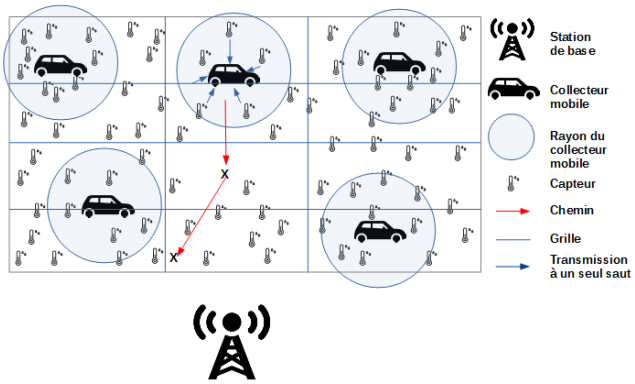


Figure 2.6 – Le protocole MQRP

2.2.4 D'autres formes de mobilité

Il existe d'autres formes de mobilité du collecteur de données développées par les chercheurs selon leurs besoins. Les plus récentes sont résumées ci-dessous.

2.2.4.1 REP (Reduced energy path), RDP (Reduced delay path) et DBP (Delay bound path)

Alsaafin et al. [50] ont proposé trois trajectoires appelées Reduced energy path (REP), Reduced delay path (RDP) et Delay bound path (DBP) pour le collecteur de données mobile. Ils ont conçu un mécanisme hiérarchique distribué d'élection de PRs basé sur la charge du nœud et son énergie résiduelle. Ensuite, ils

ont utilisé ces PRs pour déterminer une route pour le collecteur de données (Figure 2.7). Le protocole REP réduit la dissipation d'énergie globale tandis que RDP améliore la latence de REP. Le protocole DBP est approprié pour les applications intolérantes au temps. Cependant, les auteurs n'ont pas pris en compte l'urgence des paquets. Dans leur solution, les collecteurs de données visitent les PRs sans priorité.

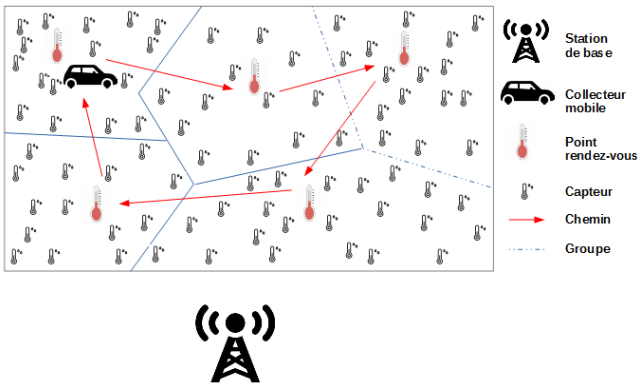


Figure 2.7 – Les protocoles REP/RDP/DBP

2.2.4.2 MDP (Markov decision processus)

Mothku et al. [51] ont proposé un processus utilisant la décision Markov qui gère les tampons pour réduire le délai de livraison des données et la consommation d'énergie dans un RCSF arborescent. Un nœud détermine la politique optimale pour réduire le nombre de retransmissions. Les tampons des capteurs sont gérés pour minimiser les paquets perdus en fonction de la prochaine période de visite du collecteur de données (Figure 2.8).

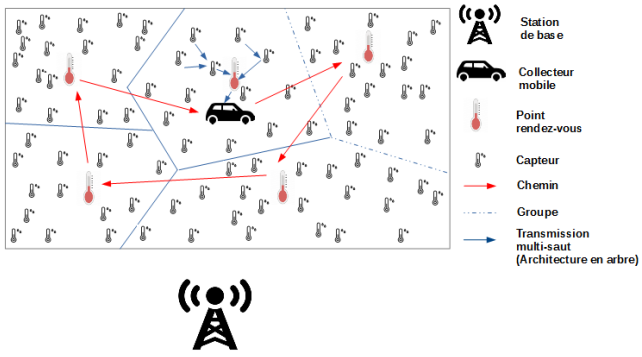


Figure 2.8 – Le protocole MDP

2.3 Limites des modèles heuristiques de mobilité

Le modèle de mobilité utilisé par le collecteur de données mobile affecte considérablement les performances et la durée de vie des nœuds de capteurs statiques et de l'ensemble du RCSF.

Parmi les limites des modèles heuristiques de mobilité, nous citons :

- Lorsque la mobilité est aléatoire
  1. La présence de nœuds surchargés dans certaines zones puisque le collecteur mobile peut visiter aléatoirement la même zone.
  2. Une diffusion fréquente des informations sur l'emplacement du collecteur de données pendant son mouvement, ce qui gaspille beaucoup d'énergie [11].
  3. Une difficulté de suivre le collecteur de données pour obtenir sa position actuelle.
  4. Une difficulté d'adopter des chemins de transmission vers le collecteur de données dans le cas de communications multi-sauts.
  5. Aucune garantie que le collecteur de données atteigne tous les nœuds du réseau ou il le fera dans un temps significatif.
- Lorsque la mobilité est géographique
  6. Ce type de mobilité est dépendant du paramètre utilisé.
- Lorsque la mobilité est prévisible
  7. Une difficulté de trouver le chemin fixe optimal en termes de durée de vie du réseau [11].

2.4 Bilan

Selon le tableau 2.1, les approches de la littérature que nous avons présenté ont utilisé un réseau homogène sans prendre en considération l'urgence des données. La latence est remarquablement diminuée lorsque le modèle de mobilité utilisé est celui proposé. La topologie utilisée joue également un rôle très important pour améliorer les performances du réseau. La durée de vie et la quantité de données collectées sont maximales quand le réseau est subdivisé en groupes.

Table 2.1 – Mobilité heuristique traitant le problème du dépassement de tampon

Protocole	**Paquets**	**Latence**	**Durée de vie**	**Topologie**	**Méthode**	**Passage à l'échelle**	**Mobilité**
DCMDC [45]	Moyen	Faible	Moyenne	Hiérarchique	Distribuée	Non	Aléatoire
CDCMP [47]	Moyen	Moyenne	Élevée	Hiérarchique	Distribuée	Non	Prédéfinie
FPM/SS [48]	Faible	Moyenne	Moyenne	Plate	/	Non	Prédéfinie
MQRP [49]	Élevé	Moyenne	Moyenne	Grille	/	Oui	Prévisible
RDP [50]	Élevé	Faible	Élevée	Hiérarchique	Distribuée	Oui	Proposée
MDP [51]	Élevé	Faible	Élevée	Arbre	/	Oui	Proposée

Dans le tableau 2.2, nous reprenons les différentes approches de regroupement heu- ristiques existantes dans la littérature et utilisant un collecteur de données mobile en précisant le type des protocoles de regroupement et le nombre de collecteurs de données utilisés dans chaque approche.

Table 2.2 – Bilan des approches de mobilité heuristiques

		Topologie					
		Arbre	Grille	Plate	Hiérarchique (distribuée)		
Collecteur de données	Unique			FPM/SS	CDCMP	Prédifinie	Modèle de mobilité
		MDP			RDP	Proposé	
	Multiple				DCMDC	Aléatoire	
			MQRP			Prévisible	

2.5 Conclusion

La topologie choisie dépend de la taille du réseau et du type d'application étudiée. L'intégration de la mobilité aux RCSFs a apporté beaucoup d'avantages pour le développement des applications dans différents domaines. Les modèles heuristiques de mobilité utilisés pour le déplacement du collecteur de données présente quelques limites affectant les performances du réseau.

Le chapitre suivant présente les algorithmes bio-inspirés utilisés pour résoudre de manière satisfaisante les défis posés par les schémas de mobilité du collecteur de données ainsi que la topologie utilisée.

Chapitre 3

La bio-inspiration dans les RCSFs mobiles

Sommaire

3.1 Approches bio-inspirées du regroupement des capteurs 22
3.2 Modèles de mobilité bio-inspirés 25
3.3 Bilan 26
3.4 Conclusion 27

Les organismes vivants ont une certaine flexibilité et robustesse leurs permettent de survivre dans des environnements complexes et dynamiques. L'interprétation des mécanismes de ces organismes du point de vue ingénierie conduira à de nouvelles méthodes applicables aux capteurs mobiles. Les algorithmes bio-inspirés ou méta-heuristiques (Figure 3.1) se subdivisent en deux classes principales : basées sur l'essaim (par exemple colonie bactérienne et colonie d'oiseaux) et basées sur l'évolution (par exemple algorithmes génétiques).

Dans cette thèse, nous nous sommes intéressés aux méthodes bio-inspirées imitant le comportement de recherche des animaux dans la nature comme la bactérie, la baleine, les loups gris et les salpes. Dans cette partie, nous présentons quelques protocoles bio-inspirés proposés dans la littérature pour traiter les problèmes de la collecte de données dans des RCSFs utilisant un collecteur de données mobile et une topologie hiérarchique. Ce chapitre vise à valoriser les caractéristiques exceptionnelles des systèmes biologiques qui les rendent une source d'inspiration bien précieuse.

3.1 Approches bio-inspirées du regroupement des capteurs

La collecte de données est aussi un problème important que la consommation d'énergie qui doit être pris en compte lors de la conception d'une topologie hiérarchique optimale. Par conséquent, la maximisation de ce paramètre conduit la conception de la topologie à un problème NP-difficile discret. Ce type de problèmes n'a pas pu être résolu en temps polynomial en utilisant des méthodes heuristiques. L'intelligence en essaim pourraient être une alternative efficace capable de trouver des solutions optimales à ce type de problèmes [5, 52]. Dans cette partie, nous présentons quelques approches de regroupements basées sur l'intelligence en essaim, plus particulièrement utilisant Grey wolf optimization (GWO) [53], Whale optimization algorithm

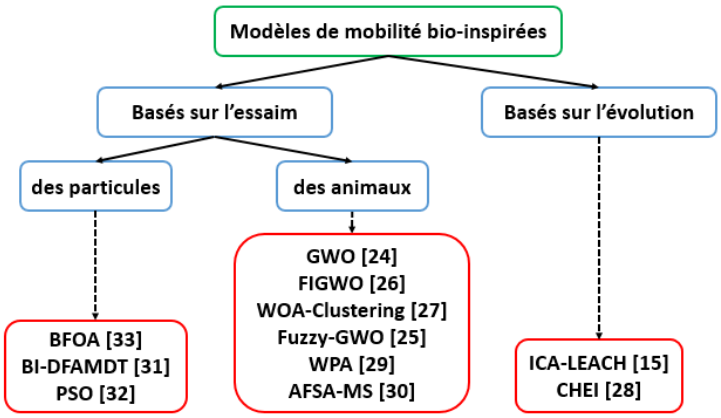


Figure 3.1 – Classification des algorithmes bio-inspirés

(WOA) [54] et Imperialistic competition algorithm (ICA).

Grey wolf optimization (GWO) et Whale optimization algorithm (WOA) ont suscité beaucoup notre attention en raison de leur simplicité, de leur facilité d'implémentation et de la diminution du nombre de paramètres de contrôle utilisés, seuls deux paramètres principaux à ajuster (a et C) dans GWO. Ils peuvent également être généralisés à des problèmes à grande échelle. De plus, ils ont une capacité spéciale à trouver le bon équilibre entre l'exploration et l'exploitation pendant la recherche, ce qui conduit à une convergence favorable. Contrairement à d'autres heuristiques basées sur la population, tels que les algorithmes génétiques, GWO enregistre les trois meilleures solutions trouvées. Imperialistic competition algorithm (ICA) a attiré notre attention par sa caractéristique notable d'être combiné avec d'autres algorithmes en aboutissant à un résultat très favorable en un temps de calcul raisonnable. En outre, il est établi sur la base d'un calcul mathématique qui nous permet d'étudier plus facilement sa convergence et sa robustesse.

3.1.1 GWO (Grey wolf optimization)

GWO [55] optimise le processus de formation de chaque groupe pour minimiser les distances intra-groupes dans un réseau homogène en s'inspirant de la hiérarchie des loups gris. La formation des groupes est conçue sur la base des nœuds chefs qui minimisent l'équation 3.1.

$$J = \sum_{j=1}^{x} \sum_{i=1}^{k} ||x_i^j - c_j||^2 \tag{3.1}$$

Où x_i^j est le nœud i du groupe j, c_j est le chef du groupe j, $(x_i^j - c_j)$ est la distance entre le nœud et le chef du groupe. Le nœud ayant l'énergie maximale restante dans le groupe est désigné chef du groupe.

3.1.2 Fuzzy-GWO

Fuzzy-GWO [56] combine l'algorithme GWO avec la logique floue pour atteindre l'efficacité énergétique lors des communications dans un réseau hétérogène. En s'inspirant de la hiérarchie des loup gris, les nœuds

calculent leurs probabilité p pour être des chefs de groupes. Cette probabilité est donnée par :

$$p = \frac{k_{opt}}{n} \tag{3.2}$$

Où k_{opt}, est le nombre optimal de chefs et n est le nombre de nœuds vivants pour le cycle actuel. La tâche principale de l'algorithme GWO est de trouver le nombre optimal des chefs de groupes afin que le nombre de nœuds vivants à chaque cycle soit maximisé. Ensuite, la logique floue est utilisée pour la formation des groupes. Lors de cette phase, chaque nœud non chef évalue trois paramètres : son énergie résiduelle, sa distance à SB et sa distance aux chefs de groupes. En se basant sur le résultat, le nœud décide de sa probabilité d'être connecté à tout chef de groupe disponible.

3.1.3 FIGWO (Fitness value based improved grey wolf optimization)

L'idée de Fitness value based improved grey wolf optimization (FIGWO) [57] est d'utiliser GWO pour le choix des chefs de groupes sélectionnés au préalable en fonction de la valeur de coût F (équation 3.3) calculée selon la distance du nœud à la SB et de son énergie résiduelle, ce qui garantit une meilleure répartition des chefs et une structure de groupe plus équilibrée.

$$F = \begin{cases} a \times \frac{E_r}{E_i} + (1-a) \times \left(\frac{d_{MAX}-d}{d_{MAX}-d_{MIN}}\right), & E_r > 0 \\ 0, & E_r \leq 0 \end{cases} \tag{3.3}$$

Où a est un cœfficient qui indique la contribution entre E_r et d dans la fonction de coût F. E_r représente l'énergie résiduelle d'un nœud vivant, tandis que E_i est l'énergie initiale d'un nœud ; d est la distance du nœud à la SB ; d_{MAX} et d_{MIN} sont respectivement la distance maximale et minimale entre un nœud et la SB. Ensuite, en fonction de la distance aux chefs et à la SB, la distance de transmission des capteurs est recalculée pour réduire la consommation d'énergie.

3.1.4 WOA-Clustering

WOA-Clustering [58] est un protocole de regroupement centralisé inspiré des baleines minimisant la dissipation d'énergie dans un réseau homogène. L'algorithme WOA, au niveau de la SB, détermine les 10 meilleures nœuds qui minimisent la fonction de coût f (équation 3.4) pour être des chefs de groupes.

$$f(CH_i) = p_1|N(CH_i)| + p_2 \sum(CH_E) \tag{3.4}$$

Où p_1 et p_2 sont choisis aléatoirement entre 0 et 1. $N(CH_i)$ est la liste de tous les nœuds voisins du nœud chef CH_i, et CH_E est le niveau d'énergie résiduelle du nœud voisin.

3.1.5 CHEI (Cluster head election using imperialistic competition algorithm)

CHEI [59] est un protocole qui se base sur l'algorithme d'impérialisme pour choisir le capteur adéquat pour être le chef du groupe. L'algorithme proposé utilise différents paramètres dans le but d'augmenter la durée de vie du réseau. Ces paramètres sont basés uniquement sur l'énergie : l'énergie résiduelle du chef,

l'énergie requise pour envoyer un message vers la SB et l'énergie requise pour recevoir un message par le chef et le transmettre à la SB. Sauf pour le premier cycle ou l'énergie de tous les capteurs est identique, ils ne considèrent donc que la distance (équation 3.5) entre les capteurs comme paramètre principal pour sélectionner les chefs.

$$D = \sum_{i=1}^{M} (DRS_i - (DRC_i + DCS)) \tag{3.5}$$

Où DRS_i représente la distance totale entre tous les nœuds et la SB, DRC_i représente la distance totale de tous les nœuds vers leurs chef et DCS représente la distance totale de tous les chefs vers la SB.

3.2 Modèles de mobilité bio-inspirés

L'utilisation de l'intelligence en essaim pour la mobilité des collecteurs de données a aidé à aboutir à de meilleures performances. Parmi les solutions trouvées dans la littérature, nous citons : WPA [60], AFSA-MS [61], BI-DFAMDT [62], PSO [63] et BFOA [64].

3.2.1 BFOA (Bacterial foraging optimization algorithm)

Ari et al. [64] ont proposé une mobilité intelligente en s'inspirant du mouvement de la bactérie Escherichia Coli. Les collecteurs de données utilisés se déplacent en se basant sur un indicateur de force du signal reçu par les capteurs.

3.2.2 WPA (Path planning of mobile sink based on wolf pack algorithm)

Jiang et al. [60] ont planifié le chemin de déplacement du collecteur mobile en utilisant l'algorithme WPA. Les auteurs ont déterminé le point rendez-vous de chaque zone de chevauchement et ont réduit la complexité de calcul. Leur fonction objective était de minimiser la longueur $L_{(T)}$ d'un chemin de déplacement constitué de plusieurs points visités T. Le collecteur de données ne visite chaque point rendez-vous qu'une seule fois par phase de collecte.

3.2.3 AFSA-MS (Artificial fish swarm algorithm for mobile sink)

Mechta et Harous [61] se sont inspirés de l'algorithme AFSA pour le mouvement du collecteur de données. Leur algorithme trouve la meilleure trajectoire en utilisant la nature organisationnelle et le comportement alimentaire des poissons dans une topologie hiérarchique centralisée. La fonction de cout utilise comme paramètre la densité des nœuds dans chaque groupe.

3.2.4 BI-DFAMDT (Backbone integrated discrete firefly algorithm for mobile data transporter)

BI-DFAMDT [62] est un algorithme heuristique discret inspiré du comportement de clignotement des lucioles proposé pour réduire la distance parcourue par le collecteur de données lors de la collecte de données dans une topologie hiérarchique. L'algorithme calcule une trajectoire optimale pour que le collecteur de données visite les chefs avec une distance de déplacement minimale en utilisant la programmation linéaire en nombres.

3.2.5 PSO (Particle swarm optimization)

Krishnan et al. [63] ont proposé un nouveau collecteur de données mobile basé sur l'algorithme PSO. Le réseau est subdivisé en plusieurs groupes. Les chefs sont choisis en fonction de l'énergie résiduelle la plus élevée. Ensuite, la SB exécute le PSO pour trouver la distance la plus courte entre tous les chefs et transmet ces informations obtenues au collecteur de données.

3.3 Bilan

Dans les deux tableaux 3.1 et 3.2, nous résumons les approches bio-inspirées du regroupement des capteurs et la mobilité bio-inspirée des collecteurs de données, respectivement, étudiées dans les deux sections précédentes. Le symbole "/" utilisé signifie qu'aucune information n'est mentionnée à propos de ces paramètres dans la littérature.

Table 3.1 – Tableau comparatif pour les approches bio-inspirées du regroupement des capteurs

Protocole	Reg. [1]	Hétéro. [2]	Équilibrage de groupes	Couverture	Latence	Durée de vie	Débit
GWO [55]	Distribué	Non	/	Élevée	/	Moyenne	Moyen
Fuzzy-GWO [56]	Distribué	Oui	/	/	/	Élevée	Élevé
FIGWO [57]	Distribué	Non	Oui	/	/	Moyenne	Moyen
ICA-LEACH [40]	Distribué	Non	/	/	/	Moyenne	/
WOA-Cluster [58]	Centralisé	Non	Oui	/	/	Faible	/
CHEI [59]	Hybride	Non	Oui	/	Faible	Moyenne	/

Il n'y a pas de regroupement qui utilise WOA ou ICA combiné avec un autre algorithme bio-inspiré appliqué à un réseau hétérogène, les seuls protocoles trouvés sont testés dans un réseau homogène. Concernant le regroupement utilisant GWO, une seule contribution est trouvée. Fuzzy-GWO [56] supporte l'hétérogénéité mais uniquement sur la base de la différence de l'énergie initiale attribuée aux capteurs. Tous ces protocoles ne prennent pas en considération le dépassement de tampon des capteurs.

La majorité des travaux proposés pour la mobilité bio-inspirée des collecteurs de données sont hiérarchiques et éco-énergétiques avec une collecte de données élevée. Par rapport aux solutions heuristiques, les solutions

1. Regroupement
2. Hétérogénéité

Table 3.2 – Tableau comparatif pour les modèles de mobilité inspirés de l'intelligence en essaim

Protocole	**Débit**	**Latence**	**Durée de vie**	**Topologie**	**Reg.**[1]	**Passage à l'échelle**	**Collecteur**
WPA [60]	/	Faible	Moyenne	Plate	/	Non	Unique
AFSA-MS [61]	Moyen	Moyenne	Élevée	Hiérarchique	Centralisée	Oui	Unique
BI-DFAMDT [62]	Élevé	Faible	Élevée	Hiérarchique	/	Oui	Unique
PSO [63]	Moyen	/	Moyenne	Hiérarchique	Distribuée	Oui	Unique
BFOA [64]	Moyen	Moyenne	Moyenne	Plate	/	Non	Multiple

bio-inspirées confèrent aux capteurs une forme d'intelligence leur offrant la capacité d'apprendre de leur environnement et de prendre des décisions lorsque cela est nécessaire. Selon le tableau 3.3, les différentes approches existantes dans la littérature présentent quelques inconvénients. Dans WPA [60], la décision de visiter une zone ou non se base sur une valeur aléatoire donc certaines zones risquent de ne pas être visitées. Dans AFSA-MS [61], l'utilisation d'un nombre d'itérations très grand fait converger AFSA-MS rapidement mais entraînera l'augmentation du temps d'exécution. BI-DFAMDT [62], le choix des chefs se pose uniquement sur le nombre de voisins à un seul saut. De plus, la méthode de regroupement n'est pas spécifiée, pour cela nous supposons que ce protocole fonctionne avec les deux techniques de regroupement. Tandis qu'à PSO [63], le choix des chefs se pose uniquement sur le niveau de l'énergie résiduelle des nœuds. Dans BFOA [64], le nombre de collecteurs de données utilisé est conséquent et coûteux (5% pour n nœuds).

Table 3.3 – Bilan des approches de mobilité bio-inspirées

<table>
<tr><td colspan="2" rowspan="3"></td><td colspan="3">Topologie</td></tr>
<tr><td rowspan="2">Plate</td><td colspan="2">Hiérarchique</td></tr>
<tr><td>Centralisée</td><td>Distribuée</td></tr>
<tr><td rowspan="3">Collecteur de données mobile</td><td>Aucun</td><td></td><td>WOA-Clustering</td><td>GWO
Fuzzy-GWO
FIGWO
ICA-LEACH
CHEI</td></tr>
<tr><td>Unique</td><td>WPA</td><td>AFSA-MS
BI-DFAMDT</td><td>PSO
BI-DFAMDT</td></tr>
<tr><td>Multiple</td><td>BFOA</td><td></td><td></td></tr>
</table>

3.4 Conclusion

Les approches bio-inspirées pour le regroupement des capteurs et la mobilité du collecteur de données que nous avons présentées dans ce chapitre sont toutes destinées à optimiser la durée de vie et le débit dans un réseau homogène et ne prennent pas en considération le problème du débordement du tampon des capteurs. En outre, certains ne sont pas applicables dans les réseaux à contraintes de temps, et ceux qui le sont ne satisfont pas certaines contraintes de collecte de données. Quelques-unes de ces méthodes seront principalement utilisées à des fins de comparaison avec les contributions présentées dans la suite de la thèse.

À partir de ces approches, nous avons remarqué que l'évaluation du calendrier de déplacement du collecteur de données est un problème important pour réduire le taux de perte de paquets. Comment les données

peuvent être conservées à l'intérieur du réseau alors que le collecteur de données n'est pas accessible. En outre, l'état du tampon (occupation) d'un capteur change dynamiquement dans le réseau. La retransmission des paquets abandonnés entraîne plus de retards et plus de consommation d'énergie dans le réseau. En particulier, les nœuds en aval sont plus susceptibles d'entraîner une surcharge du tampon que les nœuds en amont car ils doivent stocker leurs données ainsi que celles générées par les nœuds en amont jusqu'à ce que le collecteur de données passe à proximité. De plus, le coût de retransmission des paquets perdus est plus élevé pour le paquet qui a parcouru plus de sauts. Lorsqu'un paquet est abandonné, l'énergie consommée pour acheminer le paquet par les nœuds en amont est totalement gaspillée. Il est nécessaire d'estimer les limites de la consommation d'énergie et de l'occupation du tampon.

La partie suivante de la thèse présente nos différents algorithmes assurant la mobilité bio-inspirée du collecteur de données et le choix efficace des PRs dans le but de minimiser la perte de données du au débordement du tampon.

Deuxième partie

Contributions

Chapitre 4

Mouvement du collecteur de données inspiré par les bactéries

Sommaire

4.1 Bacterial foraging optimization algorithm (BFOA) 30
4.2 Mobilité du Sink basé sur BFOA (SMBFOA) 32
4.3 Conclusion 39

Dans ce chapitre, nous présentons Sink mobility based on bacterial foraging optimization algorithm (SMBFOA) [12] un nouveau modèle de mobilité du collecteur de données. Il prend en compte de nouvelles caractéristiques inspirées des bactéries Escherichia coli, assurant une meilleure collecte de données. Par conséquent, SMBFOA introduit une mobilité autonome et intelligente permettant de choisir, pour le collecteur de données, de nouvelles coordonnées qui lui assurent une position stratégique optimale, pour que les chefs de groupes puissent lui relayer le maximum de données.

4.1 Bacterial foraging optimization algorithm (BFOA)

Les activités bactériennes sont des systèmes complexes et organisés qui offrent une nouvelle approche pour résoudre des problèmes d'optimisation complexes. Le Bacterial foraging optimization algorithm (BFOA) a été introduit par Passino en 2002 [65]. Sa conception est inspirée des comportements sociaux et coopératifs intelligents des bactéries Escherichia coli. L'auteur a montré que BFOA est meilleur que PSO en termes de convergence, de robustesse et de précision.

Pour modéliser le comportement sociale de recherche de nourriture de la bactérie Escherichia coli, il a proposé quatre opérations principales dans le système d'alimentation bactérienne : "Chemotaxis", "Swarming", "Reproduction" et "Élimination et Dispersion". Dans notre conception de mobilité proposée pour les RCSFs, nous ne sommes intéressés qu'au deux premières opérations.

1. **Chemotaxis :** Ce processus vise à simuler les opérations de base qui permettent aux bactéries Escherichia de se déplacer via les flagelles [66]. L'étape chimiotactique correspond à une nage, un

mouvement sans changer la direction [67], suivi d'une nage ou d'une rotation, modification aléatoire de la direction du mouvement [67].

Soit j, k et l les indices de l'étape de chimiotaxie, de la reproduction et de l'événement d'élimination-dispersion respectivement, tels que : $\theta^i(j,k,l) \in \mathbb{R}, i = 1,2,...,S$ définit la position de chaque membre dans la population de bactéries S au j^{ème} stade chimiotactique, au k^{ème} stade de reproduction et à l'événement l^{ème} d'élimination-dispersion. $C(i) > 0$ est la taille du pas effectué dans la direction aléatoire spécifiée par la rotation. Pour représenter une rotation, un vecteur unitaire aléatoire de la direction, disons $\phi(i)$, est générée selon l'équation 4.1.

$$\phi(i) = (\Delta^T(i)\Delta(i))^{-1/2} \times \Delta(i) \tag{4.1}$$

Où $\Delta(i) \in \mathbb{R}^p$ est un vecteur généré aléatoirement, où chaque élément de $\Delta(i)$, $i = 1,2,..,p$ est un nombre pris entre $[-1,1]$. Liu et Passino [68] ont modélisé la chimiotaxie, c'est-à-dire le mouvement de la bactérie, par l'équation 4.2.

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i) \times \phi(i) \tag{4.2}$$

2. Swarming : Ce processus consiste à former un groupe de bactéries de manière auto-organisée. Dans l'équation 4.3, la fonction $J_{cc}(\theta, \theta^i(j,k,l))$, qui modélise la signalisation de cellule à cellule via un attractif et un répulsif, est donnée.

$$\begin{aligned} J_{cc}(\theta, \theta^i(j,k,l)) = &-d_{attractif} \times \exp(-w_{attractif} \times \sum_{m=1}^{p} (\theta_m - {\theta^i}_m)^2) \\ &+ h_{répulsif} \times \exp(-w_{répulsif} \times \sum_{m=1}^{p} (\theta_m - {\theta^i}_m)^2) \end{aligned} \tag{4.3}$$

Où $d_{attractif}$ est la profondeur de l'attracteur libéré par la cellule, $w_{attractif}$ est la largeur du signal attractif, $h_{répulsif} = d_{attractif}$ est la hauteur de l'effet répulsif (magnitude), et $w_{répulsif}$ est la largeur du répulsif.
La fonction objective représentant la signalisation de cellule à cellule dans l'essaim d'Escherichia coli est définie par l'équation 4.4.

$$J_{cc}(\theta, P(j,k,l)) = \sum_{i=1}^{S} J_{cc}(\theta, \theta^i(j,k,l)) \tag{4.4}$$

Où $P(j,k,l) = \theta^i(j,k,l)$ avec $i = 1,2,...,S$ est la position de chaque membre dans la population de bactéries S dans la j^{ème} étape chimiotactique, k^{ème} étape de reproduction et l^{ème} événement d'élimination-dispersion.
L'équation 4.5 représente la fonction de coût à l'emplacement $\theta^i(j,k,l)$ de la i^{ème} bactérie.

$$J(i,j,k,l) = J(i,j,k,l) + J_{cc}(\theta^i(j,k,l), P(j,k,l)) \tag{4.5}$$

4.2 Mobilité du Sink basé sur BFOA (SMBFOA)

Un organigramme de notre première proposition est représenté par la figure 4.1. Le protocole se déroule en "cycles" qui ont approximativement le même intervalle de temps prédéterminé. Chaque cycle commence par une phase de regroupement, où se trouve l'emplacement optimal des chefs. Il est suivi d'une phase de routage, où les données intra-groupes captées sont transférées aux chefs comme dans le protocole Clustering duty cycle mobility aware protocol (CDCMP) [47]. Après cela, le collecteur de données se déplace intelligemment inspiré par le BFOA pour trouver la meilleure position pour recevoir plus de données. C'est de cette manière que se font les transmissions inter-groupes.

4.2.1 Phase de regroupement et élection des chefs

Cette phase consiste à diviser le réseau en quatre régions, chacune contenant un chef, afin d'assurer une répartition équitable des chefs dans le réseau [47]. La sélection de ces derniers se fait en fonction de la densité topologique des nœuds (D_t) lors du cycle 0 et en fonction de l'énergie résiduelle des capteurs lors des autres cycles. La densité de chaque nœud S est calculée en utilisant la formule 4.6.

$$D_t(S) = \frac{(D + L)}{D} \tag{4.6}$$

Ou D est le degré du nœud S et L est le lien entre les voisins du nœud S. Chaque chef agit en tant que coordinateur local pour coordonner la transmission des données au sein de son groupe. Il divise le groupe en deux sous-niveaux (Niveau 1 et Niveau 2) pour créer une planification Time-division multiple access (TDMA) commençant par les nœuds les plus éloignés. Il attribue en même temps, à chaque nœud membre, un intervalle de temps pendant lequel il peut transmettre ses données. Le chef réalise également la notion du Duty-Cycle pour synchroniser le temps de réveil et de sommeil dans le réseau.

4.2.2 Phase de routage

La collecte des données dans SMBFOA se fait en deux étapes. Tout d'abord, les données captées sont transférées vers les chefs (transmissions intra-groupes). Ensuite, chaque chef agrège les données reçues de ses membres pour les envoyer au collecteur de données (transmissions inter-groupes). En attendant, le collecteur de données choisit sa meilleure position selon l'algorithme BFOA.

4.2.2.1 Transmissions intra-groupes

Les auteurs de la référence [47] ont subdivisé cette étape en plusieurs périodes. Chaque période contient un nombre d'emplacements égal au nombre de membres du groupe défini. Un slot est dédié à un seul nœud membre d'un groupe, pendant lequel il transmet ses données au niveau concerné. Ces opérations sont répétées à chaque période jusqu'à la fin du cycle.

4.2.2.2 Transmissions inter-groupes

A la fin de chaque période, le chef rassemble les données de ses membres et les envoie au collecteur de données.

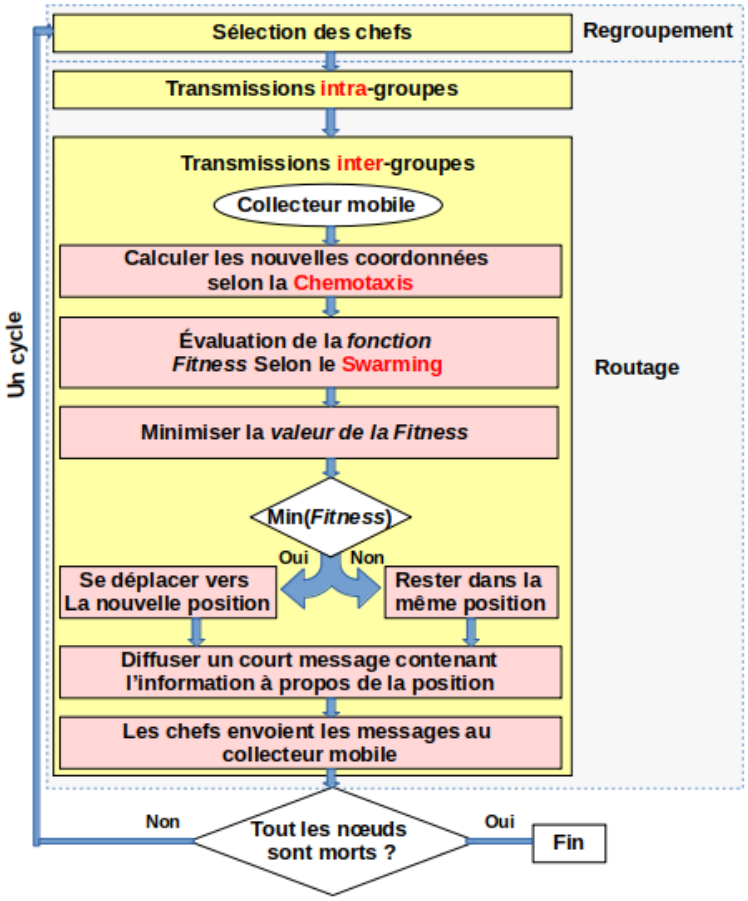


Figure 4.1 – Organigramme du protocole SMBFOA

1. Le collecteur de données se déplace selon la "Chemotaxis" : Le collecteur de données change son emplacement comme une bactérie, de la position $\theta(t)$ à $\theta(t+1)$ comme suit :

$$\begin{aligned} \theta(t+1) &= \theta(t) + C \times \phi \\ \phi &= (\Delta^T \times \Delta)^{-1/2} \times \Delta \times Degrée \end{aligned} \tag{4.7}$$

ϕ : est une direction aléatoire de longueur unitaire.

C : est la taille du pas effectué dans la direction aléatoire spécifiée par ϕ. Elle doit être positive $C > 0$.

Δ : est un vecteur de deux éléments Δ_1 et Δ_2, qui sont choisis au hasard dans l'intervalle [-1,1] comme mentionné dans la référence [65].

$Degrée$: est le nombre de membres du groupe le plus dense du réseau. Le collecteur de données se déplace vers ce groupe afin de collecter autant de données que possible.

$$\Delta \times \Delta^T = \sum_{i=1}^{n} {\Delta_i}^2 \tag{4.8}$$

$\Delta \times \Delta^T$: est le carré scalaire d'un vecteur, i.e le produit d'un vecteur par sa transposée.

2. **Le collecteur de donnée évalue la fonction de coût selon "Swarming" :** Cette évaluation se fait en utilisant la formule 4.9.

$$J(\theta^k(t+1)) = -\alpha_1 \times g(\beta_1) + \alpha_2 \times g(\beta_2) \tag{4.9}$$

Où α_1, α_2, β_1 et β_2 sont des cœfficients qui doivent être choisis correctement. Soit $g(x)$ la fonction swarming qui régit le mouvement du collecteur de données.

$$\begin{aligned} g(x) &= \exp\left(-x \times \Gamma(\theta^k(t+1))\right)\Gamma(\theta^k(t+1)) \\ &= (\theta_1(t+1) - \theta_1(t))^2 + (\theta_2(t+1) - \theta_2(t))^2 \end{aligned} \tag{4.10}$$

Pour minimiser la fonction de coût, nous devons vérifier l'équation 4.11.

$$J(\theta^K(t)) > J(\theta^K(t) + \Delta\theta^k) \tag{4.11}$$

4.2.3 Exemple

Nous supposons que les coordonnées de notre collecteur de données sont (50 ;50). Il va se déplacer vers de nouvelles positions selon BFOA.

1. **Le collecteur de données se déplace selon la "Chemotaxis" :** Nous prenons deux nombres aléatoires entre [-1 ;1]. Par exemple : (-1 ;1), (0.5 ;0.5), (0.9 ;-0.5), (0.75 ;-0.75).

Tout d'abord, nous calculons le carré scalaire :

$$\Delta \times \Delta^T = [-1, 1] \times [-1, 1]^T = 2$$

Ensuite, nous calculons :

$$\phi = (\Delta \times \Delta^T)^{-1/2} \times \Delta \times Degrée) = 2^{-1/2} \times [-1, 1] \times 8 = [-5.6568, 5.6568]$$

Après, les coordonnées de la nouvelle position (X, Y) sont calculées avec $C = 0.1$:

$$\theta_x(t+1) = 50 + (-5.6568) \times 0.1 = 49.4343 \text{ et } \theta_y(t+1) = 50 + (5.6568) \times 0.1 = 50.5656$$

Pour chaque nouvelle position, la fonction de coût doit être évaluée.

2. **Le collecteur de données évalue son coût selon le "Swarming" :** Nous devons soigneusement choisir la valeur des paramètres suivants : α_1 , α_2 , β_1 et β_2.

$$J(\theta^k(t+1)) = -0.5 \times g(0.1) + 1.5 \times g(0.1)$$

Si $J(\theta^K(t+1)) < J(\theta^K(t)$, alors le collecteur de données se déplacera vers la nouvelle position car il minimisera la fonction de coût. Il y aura donc moins de mouvements. Il s'assurera que son mouvement collectera autant de données que possible sans déplacement inutile.

La figure 4.2 montre le mouvement du collecteur de données dans l'exemple en utilisant le BFOA.

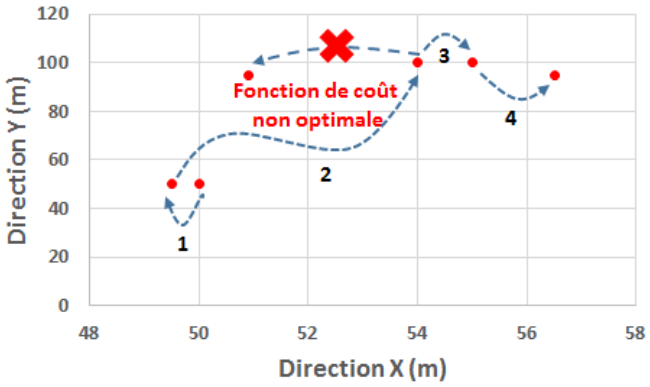


Figure 4.2 – Positions du collecteur de données avec la mobilité BFOA

4.2.4 Évaluation des performances de SMBFOA

Dans cette section, nous effectuons des simulations à l'aide de Network simulator (NS) 2 pour analyser et évaluer les performances du protocole proposé. Le nombre de nœuds dans le réseau simulé est de 100, dispersés aléatoirement dans une zone de $100 \times 100\ m^2$ comme le montre la figure 4.3.

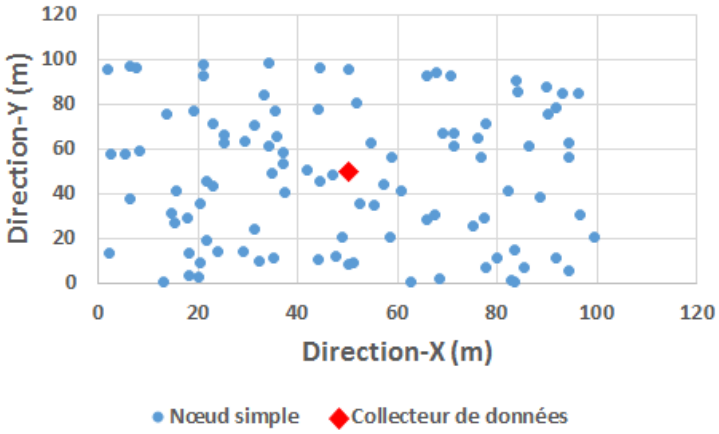


Figure 4.3 – Déploiement des nœuds

Les paramètres de la simulation sont présentés dans le tableau 4.1. Les hypothèses suivantes concernant le modèle de réseau sont fixées :

- Le collecteur de données est un appareil riche en ressources et n'est pas limité en termes de mémoire et de puissance de calcul.
- Il se situe à l'intérieur du réseau.
- Tous les nœuds capteurs sont stationnaires après le déploiement.
- Tous les capteurs sont équipés d'un Global position system (GPS) ou d'autres appareils de localisation.

- Le réseau comprend des nœuds homogènes.
- Initialement, les nœuds capteurs ont la même quantité d'énergie.

Table 4.1 – Paramètres de simulation

Paramètres	Valeurs
Taille du réseau (m^2)	100 × 100
Localisation du collecteur	(50,50)
Nombre de nœuds	100
Position des nœuds	entre (0,0) et (100,100)
Taille du paquet (bits)	2000
Temps de simulation (s)	3600
Nombre de groupes	4
Énergie initiale des nœuds (J)	2
Temps du cycle (s)	20

La figure 4.4 représente l'évaluation des performances des protocoles CDCMP et SMBFOA en termes de nombre de paquets de données reçus par le collecteur de données et l'énergie consommée par les nœuds. Comme le montre la figure 4.4, la communication inter-groupes minimise la consommation d'énergie lors de la transmission de données vers le collecteur. Ces résultats ont révélé que, pour le réseau utilisant le protocole CDCMP, ou le collecteur de données suit un chemin rectangulaire, 70475 paquets de données ont été transmis en consommant 160 J à l'instant 460 s. En revanche, pour un réseau utilisant le protocole SMBFOA, le collecteur de données reçoit 99141 paquets de données en utilisant seulement 155 J. En d'autres termes, lorsque les $\frac{3}{4}$ de l'énergie d'un réseau est consommée, le protocole CDCMP ne transmet que $\frac{2}{3}$ des paquets de données par rapport au protocole SMBFOA. Nous notons une économie d'énergie dans le cas où la mobilité intelligente est intégrée dans le protocole SMBFOA. Si le collecteur de données reste immobile pendant une longue période, les chefs éloignés de celui-ci consommeront plus d'énergie et leur durée de vie diminuera rapidement, comme dans le cas du protocole CDCMP.

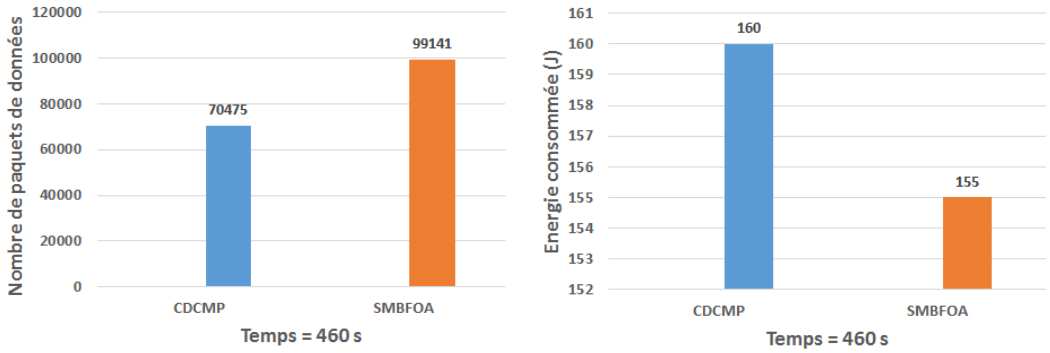


Figure 4.4 – Nombre de paquets reçus par le collecteur de données et consommation énergétique à l'instant T = 460 s de SMBFOA vs CDCMP

La figure 4.5 illustre l'évaluation des performances des protocoles CDCMP et SMBFOA en termes de nombre de paquets de données reçus par le collecteur de données. La quantité de paquets de données reçus dans un

certain laps de temps est un indice important pour mesurer la qualité du service du réseau. Les résultats de la simulation montrent clairement que le protocole SMBFOA est meilleur que le protocole CDCMP en termes de nombre de paquets de données reçus par le collecteur de données. En utilisant le protocole SMBFOA ce nombre est pratiquement doublé, 16×10^4 paquets de données reçus comparé à son homologue, le protocole CDCMP, qui reçoit que 10×10^4 paquets de données. Suite à l'utilisation idéale de la synchronisation de l'horloge et de la fonction de coût, nous avons obtenu une amélioration significative de la quantité de données reçue. La raison est simple, le mouvement intelligent du collecteur de données lui permet de se rapprocher de plus en plus vers les chefs. Ainsi, la distance inter-groupes sera minimisée. Les chefs consomment moins d'énergie et transmettent plus de données.

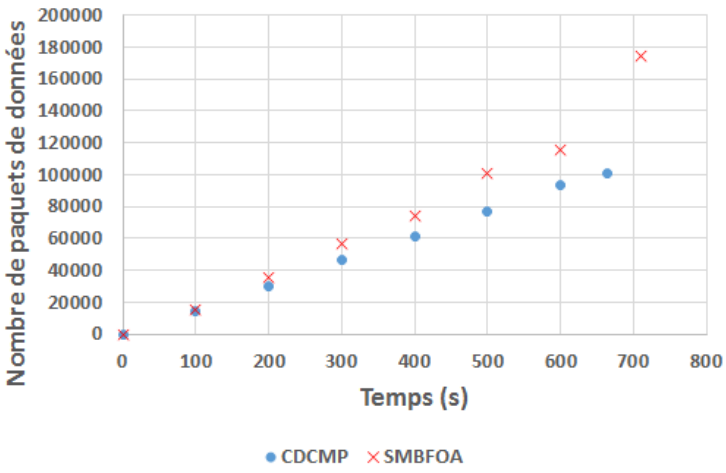


Figure 4.5 – Nombre de paquets de données reçus par le collecteur de données de SMBFOA vs CDCMP

La figure 4.6 montre l'évaluation des performances des protocoles CDCMP et SMBFOA en termes de durée de vie. Nous constatons que le premier nœud est mort après 470 s et tous les nœuds sont morts après 680 s dans le protocole CDCMP. Cependant, dans le protocole SMBFOA, le premier nœud est mort également après 470 s et tous les nœuds sont morts après 710 s. Il est clair que le protocole proposé ne diffère pas trop du protocole CDCMP. Cela est lié au fait que les deux protocoles utilisent la même méthode de regroupement et de routage intra-groupes. De plus, la consommation d'énergie est répartie entre les chefs grâce à l'utilisation d'un collecteur de données mobile intelligent. En conséquence, aucun nœud ne sera exploité plus que sa capacité, ce qui à son tour conduit à l'optimisation de la consommation d'énergie dans l'ensemble du réseau.

La figure 4.7 illustre l'évaluation des performances des protocoles CDCMP et SMBFOA par le rapport à la consommation d'énergie et le débit. On peut voir qu'après 150 J consommés, le protocole CDCMP transmet seulement 65000 paquets de données contrairement au protocole SMBFOA qui transmet 90000 paquets de données.

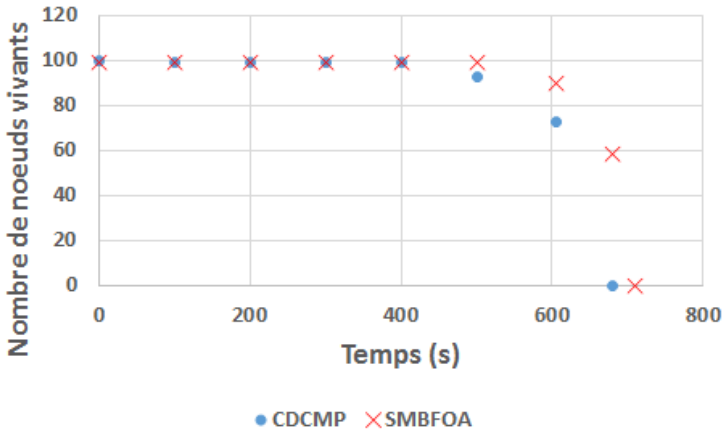


Figure 4.6 – Durée de vie de SMBFOA vs CDCMP

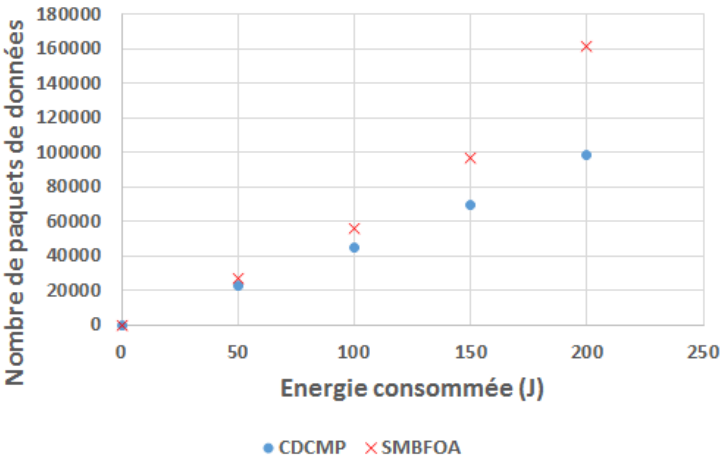


Figure 4.7 – Consommation énergétique de SMBFOA vs CDCMP

4.3 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle approche de mobilité bio-inspirée nommée Sink mobility based on bacterial foraging optimization algorithm. Elle est une approche de routage inter-groupe économe en énergie et augmente la quantité de paquets de données collectées. SMBFOA s'inspire du mouvement de la bactérie Escherichia coli pour le choix du chemin à emprunter lors de la collecte de données dans le réseau.

Les résultats de la simulation montrent que les performances de SMBFOA sont meilleures que celles de CDCMP en termes de consommation d'énergie et de débit. Le collecteur de données à mobilité intelligente utilisé dans SMBFOA contribue à conserver l'énergie globale du système avec une opérabilité maximale. Cela équilibre la durée de vie des chefs et augmente considérablement la quantité de données collectées.

L'approche que nous avons proposée dans ce chapitre s'appuie sur un algorithme de regroupement et de routage se basant sur des graphes modélisant des réseaux de topologie homogène. Cependant, les réseaux IdO constituent principalement de capteurs ayant différents caractéristiques (batterie et mémoire distinctes), et interagissent, au cours de leurs fonctionnement, avec divers autres capteurs. Dans ce cas, nous traitons avec des capteurs qui évoluent au sein de réseaux de topologie hétérogène. L'impact de ce type de topologie sur notre algorithme doit être analysé.

De plus, la méthode de regroupement utilisée, est la même que celle de CDCMP. Elle se base uniquement sur un seul paramètre qui est l'énergie résiduelle du nœud. La disparition d'un nœud du réseau, pour des raisons autres que la faible énergie, a des conséquences inévitables sur la connectivité et la couverture de la zone surveillée, et ce, sur la quantité de données collectées. Cependant, d'autres paramètres doivent être pris en compte lorsque la topologie du réseau est hétérogène.

Chapitre 5

Regroupement inspiré par les loups gris et les baleines

Sommaire

5.1 Solutions bio-inspirées 40
5.2 Solutions bio-inspirées hybrides pour le débordement du tampon au niveau des nœuds non chefs 42
5.3 Exemple 52
5.4 Modélisation et vérification 53
5.5 Modèles 55
5.6 Évaluation des performances de GWO et WOA 57
5.7 Exemple d'application 63
5.8 Conclusion 66

Les RCSFs hétérogènes présentent de nouvelles contraintes lors de la collecte de données dans les applications en temps-réel, surtout si la différence entre la taille du tampon des capteurs et très importante. Cela influencera constamment sur la quantité de données collectées.

Dans ce chapitre, nous nous intéressons au problème du débordement du tampon des capteurs dans un réseau hétérogène sans l'utilisation du collecteur de données. Il s'agit de construire une nouvelle méthode de regroupement hiérarchique de nœuds, appelée GWO/WOA [69], afin que, en prenant en compte une nouvelle caractéristique (la taille du tampon du capteur), le nombre de paquets perdus soit minimales. Notre objectif est de construire des groupes équilibrés, en combinant la force de l'algorithme ICA avec les algorithmes GWO et WOA. Le protocole imite le processus hiérarchique des loups gris et le comportement de chasse des baleines dans la nature, ce qui induit une formation autonome et intelligente des groupes.

5.1 Solutions bio-inspirées

Au cours des dernières années, de nombreuses recherches ont été publiées dans le domaine des algorithmes bio-inspirés. Leurs résultats sont optimaux lorsque plusieurs techniques de ce type d'algorithmes sont combi-

nées. Cependant, dans la plupart des références bibliographiques que nous avons examinées, chaque heuristique est implémentée seule sans être combinée à d'autres algorithmes heuristiques. L'algorithme compétitif impérialiste ICA peut être hybridé avec certains algorithmes bien connus tels que l'algorithme d'optimisation des groupes de loups GWO et l'algorithme d'optimisation des groupes de baleines WOA. Cette hybridation peut être mise en œuvre de différentes manières. Par exemple, GWO ou WOA peuvent être utilisés comme point de départ pour générer une bonne solution initiale et le reste de la recherche peut être pris en charge par l'ICA. Dans certains cas, l'ICA peut être utilisé comme point de départ pour générer une solution initiale et d'autres algorithmes peuvent être utilisés pour effectuer la recherche. En outre, GWO et WOA peuvent être combinés à l'ICA en tant qu'outil complémentaire pour promouvoir la capacité d'exploitation et produire des solutions de haute qualité. GWO ou WOA peuvent être utilisés pour sélectionner les chefs des groupes et ICA pour former les groupes. Dans notre solution, la dernière hypothèse est retenue.

5.1.1 GWO

En 2014, Mirjalili et al. [70] ont proposé l'algorithme GWO, une méthode qui imite la hiérarchie de leadership et le processus de chasse des loups gris dans la nature. Cette hiérarchie emploie quatre sortes de loups : alpha, bêta, delta et oméga [71]. Alpha est le leader, il est chargé de prendre des décisions. Il est également appelé le loup dominant, car il doit être suivi par la meute. C'est le meilleur membre pour gérer le pack, mais ce n'est pas automatiquement le loup le plus fort. Les loups bêta sont au deuxième niveau de la hiérarchie. Ils aident les loups alpha à prendre des décisions mais doivent les respecter et commandent les loups du niveau inférieur. Dans le cas où l'alpha décède, les bêtas sont les candidats appropriés pour le remplacer. Omega est le loup gris le moins bien classé. Cette catégorie de loups doit appliquer toutes les décisions des autres loups dominants. Si un loup n'est pas un alpha, un bêta ou un oméga, il est appelé delta (ou subordonné dans certaines références). Les loups de ce niveau dominent les omégas mais doivent se soumettre aux alphas et aux bêtas. Les loups gris ont un autre comportement social intéressant en plus de la hiérarchie sociale. Ce comportement est la chasse en groupe. Les principales phases de cette technique sont les suivantes : rechercher des proies, les encercler et les attaquer par la suite [72].

5.1.2 WOA

WOA est un algorithme inspiré de la vie des baleines, introduit par Mirjalili et al. [73]. Ces poissons ont une méthode de chasse unique appelée méthode d'alimentation par filet à bulles. Cela implique généralement de créer des bulles le long d'un cercle autour de la proie.

L'algorithme WOA commence avec une population de baleines (solutions) générée aléatoirement, chacune avec une position aléatoire. Dans la première itération, les agents de recherche mettent à jour leurs positions en se référant à un agent de recherche choisi au hasard. Cependant, à partir de la deuxième itération, les agents de recherche mettent à jour leurs positions par rapport à la meilleure solution obtenue jusqu'à présent. Un agent de recherche aléatoire est choisi si la valeur de $|A| > 1$, cela aide à l'exploration. Lorsque la meilleure solution est sélectionnée, $|A|$ est défini sur $|A| < 1$. Cela induit une exploitation car tous les agents de recherche vont converger. Le comportement de chasse peut être expliqué en 3 phases : chercher, encercler et attaquer la proie. Les deux premières étapes ne diffèrent pas de l'algorithme GWO. La différence réside uniquement dans la façon d'attaquer la proie.

5.1.3 ICA

ICA est un algorithme qui met le contrôle sur de nombreux pays et utilise leurs sources une fois que les colonies sont dominées par des règles. Il a été proposé en 2007 par Atashpaz et Lucas [74]. Différemment des étapes de déroulement de la plupart des algorithmes méta-heuristiques inspirés d'un phénomène évolutif basé sur la nature, ce processus est appelé le processus socio-politique de l'impérialisme.

5.2 Solutions bio-inspirées hybrides pour le débordement du tampon au niveau des nœuds non chefs

5.2.1 La sélection des chefs basée sur GWO

Dans cette étude, la sélection des chefs est faite sur un réseau hétérogène et inspirée par la structure du loup gris. Le tableau 5.1 présente la similitude et la comparaison entre le loup gris et le RCSF.

.

Table 5.1 – Similitude entre GWO et le RCSF

GWO	**RCSF**
Loups gris	Nœuds de capteur
Emplacement des aliments (proies)	Fonction objective
La position des proies dans la région n'est pas connue par les loups gris.	Les nœuds ne connaissent pas à l'avance la fonction de coût. Les nœuds ne peuvent détecter que les modifications des valeurs de cette fonction.
Meilleure valeur de la fonction de coût	Nœud chef
Le nombre moyen de loups par meute se situe entre 5 et 12.	Le nombre idéal de membres dans un groupe est inférieur à 12 et supérieur à 5.
Positions des loups	Positions des nœuds (pos(t)=x(t),y(t))

Nous expliquerons dans ce qui suit comment sélectionner les chefs à l'aide de GWO en utilisant l'algorithme 5.1.

5.2.1.1 Recherche de proies (Exploration)

Conformément aux positions des loups alphas, bêtas et deltas, les loups gris recherchent principalement les proies. Ils convergent les uns des autres pour attaquer les proies et divergent pour les rechercher. Afin de modéliser mathématiquement la divergence, nous utilisons le paramètre A avec des valeurs aléatoires supérieures à 1 ou inférieures à -1 pour obliger l'agent de recherche de diverger de la proie. La valeur de ce paramètre est contrôlée par a, qui diminue linéairement de 2 à 0. C'est un autre paramètre de GWO qui favorise l'exploration. Il contient des valeurs aléatoires comprises dans l'intervalle $[0; 2]$. Cette contribution

Algorithme 5.1 : *Élection des chefs en utilisant GWO*

Entrées : Réseau de N nœuds
Sorties : Les nœuds $Chefs$
- Initialisez la population de loups gris $X_i(i = 1, 2, ..., n)$
- Initialisez a, A et C
- Calculez le $coût$ de chaque agent
- Choisir trois meilleures solutions X_{alpha}, X_{beta} et X_{delta}

Tant que *itération* $<$ *nombre max d'itérations* **Faire**
  Pour chaque agent de recherche **Faire**
    - Mettre à jour la position de l'agent de recherche actuel
    Si cycle = 0 **Alors**
      - Utilisez les formules 5.7, 5.8 et 5.9
    Sinon
      - Utilisez les cas 1, 2, 3 ou 4
    Fin Si
  Fin Pour
  - Mettre à jour a, A et C
  - Calculez le $coût$ de tous les agents de recherche
  - Mettre à jour X_{alpha}, X_{beta} et X_{delta}
  - itération = itération + 1
Fin Tant que
Retournez Les nœuds Chefs

est forte lorsque $C < 1$, la solution gravite davantage vers les proies, favorisant l'exploration et évitant les optima locaux [72].

5.2.1.2 Encerclement de la proie

Comme mentionné ci-dessus, la proie est encerclée pendant la chasse par des loups gris. Pour modéliser mathématiquement ce comportement, les équations suivantes sont proposées :

$$X(t+1) = X_p(t) - A \times D \tag{5.1}$$

Où $X(t+1)$ est l'emplacement suivant du loup, $X(t)$ est l'emplacement actuel, A est une matrice de cœfficients et D est un vecteur qui dépend de l'emplacement de la proie X_p et est calculé comme suit :

$$D = \left|C \times X_p(t) - X(t)\right| \tag{5.2}$$

Où

$$C = 2 \times r_2 \tag{5.3}$$

r_2 est un vecteur généré aléatoirement à partir de l'intervalle $[0; 1]$. Notez que ces équations utilisent des vecteurs, donc cela s'applique à n'importe quel nombre de dimensions. Notez que les composants aléatoires dans les équations ci-dessus simulent différentes tailles de pas et vitesses de déplacement des loups gris. Les équations pour définir leurs valeurs sont les suivantes :

$$A = 2 \times a \times r_1 - a \tag{5.4}$$

Où a est un vecteur dont les valeurs sont diminuées linéairement de 2 à 0 au cours de l'exécution. r_1 est un vecteur généré aléatoirement à partir de l'intervalle $[0; 1]$. L'équation pour mettre à jour le paramètre a est la suivante :

$$a = 2 - t \times (\frac{2}{T}) \tag{5.5}$$

Où t montre l'itération actuelle et T est le nombre maximal d'itérations [72].

Une version modifiée de GWO a été proposée par Mittal et al. [75] pour profiter d'une meilleure exploration. Au lieu de diminuer la valeur d'une manière linéaire, les auteurs ont utilisé une fonction exponentielle donnée par :

$$a = 2 \times \left(1 - (\frac{t^2}{T^2})\right) \tag{5.6}$$

5.2.1.3 Chasse (Optimisation)

Dans GWO, il est évident que alpha, bêta et delta sont toujours les trois meilleures solutions obtenues jusqu'à présent. L'optimum global dans les problèmes d'optimisation est inconnu, il a donc été supposé que alpha, bêta et delta ont une bonne idée de son emplacement, ce qui est raisonnable car ce sont les meilleures solutions dans l'ensemble de la population [72]. Par conséquent, les autres loups devraient être obligés de mettre à jour leurs positions. Pour le premier cycle, les positions sont mises à jour comme suit :

$$\begin{aligned} X_1 &= X_{alpha}(t) - A_1 \times D_{alpha} \\ X_2 &= X_{beta}(t) - A_2 \times D_{beta} \\ X_3 &= X_{delta}(t) - A_3 \times D_{delta} \end{aligned} \tag{5.7}$$

Où D_{alpha}, D_{beta} et D_{delta} sont calculés en utilisant la formule 5.8 comme suit :

$$\begin{aligned} D_{alpha} &= \left|C_1 \times X_{alpha} - X\right| \\ D_{beta} &= \left|C_2 \times X_{beta} - X\right| \\ D_{delta} &= \left|C_3 \times X_{delta} - X\right| \end{aligned} \tag{5.8}$$

$X(t+1)$ est obtenue en utilisant la formule 5.9.

$$X(t+1) = \frac{1}{3}X_1 + \frac{1}{3}X_2 + \frac{1}{3}X_3 \tag{5.9}$$

Pour les autres cycles, les positions sont mises à jour en utilisant l'un des cas suivants :
- cas 1 :

$$\begin{aligned} D_{alpha} &= \left|C_1 \times Coût_{alpha} - Coût_X\right| \\ D_{beta} &= \left|C_2 \times Coût_{beta} - Coût_X\right| \\ D_{delta} &= \left|C_3 \times Coût_{delta} - Coût_X\right| \\ X_1 &= Coût_{alpha} - A_1 \times D_{alpha} \\ X_2 &= Coût_{beta} - A_2 \times D_{beta} \\ X_3 &= Coût_{delta} - A_3 \times D_{delta} \\ X(t+1) &= \frac{1}{3}X_1 + \frac{1}{3}X_2 + \frac{1}{3}X_3 \end{aligned} \tag{5.10}$$

- cas 2 :

$$\begin{aligned} X_1 &= \left|Coût_{alpha} - Coût_X\right| \\ X_2 &= \left|Coût_{beta} - Coût_X\right| \\ X_3 &= \left|Coût_{delta} - Coût_X\right| \\ X(t+1) &= \frac{1}{3}X_1 + \frac{1}{3}X_2 + \frac{1}{3}X_3 \end{aligned} \tag{5.11}$$

- cas 3 :

Une autre version modifiée de GWO a été proposée dans la référence [76]. Les auteurs visaient à avoir une convergence plus rapide en mettant à jour la position des loups proportionnellement à l'itération actuelle, comme indiqué :

$$X(t+1) = \frac{1}{iteration}^{\frac{Coût_{alpha} - Coût_X}{Coût_{alpha} - Coût_{pire}}} \tag{5.12}$$

- cas 4 :

$$X(t+1) = \left|\frac{Coût_{alpha} - Coût_X}{Coût_{alpha} - Coût_{pire}}\right| \tag{5.13}$$

5.2.1.4 Attaque de proies (Exploitation)

Comme mentionné ci-dessus, les loups gris terminent la chasse en attaquant la proie lorsqu'elle cesse de se déplacer. L'exploitation est favorisée lorsque $-1 < A < 1$. Un bon équilibre entre exploitation et exploration est nécessaire pour trouver une approximation précise de l'optimum global à l'aide d'algorithmes stochastiques. Cet équilibre est atteint par GWO avec le comportement décroissant du paramètre a dans l'équation du paramètre A [72]. Avec une baisse de A, la moitié de la recherche est consacrée à l'exploration ($A \geq 1$) et l'autre moitié est consacrée à l'exploitation ($A < 1$), comme le montre la figure 5.1.

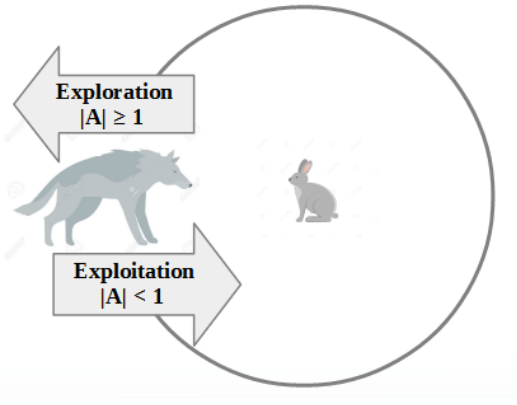


Figure 5.1 – Période d'exploration par rapport à la période d'exploitation en fonction de la valeur de A dans GWO

5.2.1.5 Condition de terminaison

Elle dépend de deux modes : si le nombre d'itérations est spécifié, la convergence vers une réponse optimale quand on n'est pas au courant de la valeur optimale, s'il n'y a pas de changement après un certain nombre d'itérations [77]. Dans ce travail, le critère d'arrêt se fait par le nombre d'itérations.

Notre objectif est de maximiser la fonction de coût décrite ci-dessus, en choisissant le nœud ayant plus d'énergie, plus d'espace mémoire et plus de voisinage. Current battery power (CBP) représente l'énergie résiduelle.

$$\begin{aligned} Coût(i) =& coef_1 \times NombreDeVoisins \\ &+ coef_2 \times CBP(i) \\ &+ coef_3 \times Capacité(i) + Pertinence(i) \end{aligned} \tag{5.14}$$

Oû :

$$coef_1 = 0.2 \qquad coef_2 = 0.3 \qquad coef_3 = 0.5$$

Ces cœfficients sont choisis par priorité. Cela signifie que nous privilégions davantage l'espace mémoire du nœud ($coef_3$), puis son énergie résiduelle ($coef_2$), ensuite le voisinage ($coef_1$). Il s'agit de résoudre le problème de dépassement de mémoire tampon et de minimiser autant que possible la perte de données.
Dans ce qui suit, nous expliquerons en détail les paramètres de la fonction de coût. L'emplacement du nœud est calculé comme dans la formule 5.15. Où $DistMaxSB$ et $DistMinSB$ sont formulés à partir des équations 5.16 et 5.17 respectivement. Les coordonnées $(X_{min}; Y_{min}) = (100; 100)$ et $(X_{max}; Y_{max}) = (0; 0)$ représentent respectivement les points les plus proches (P_P) et les plus éloignés (P_E) de la SB.

$$Localisation(i) = \begin{cases} \textit{Si i = } P_E \textit{ ou } P_P & 1 \\ \textit{autres situations} & \frac{DistMaxSB - Dist(i)ToBS}{DistMaxSB - DistMinSB} \end{cases} \tag{5.15}$$

$$DistMaxSB = \max\left(\sqrt{\left(X_{min} - X_{SB}\right)^2 + \left(Y_{min} - Y_{SB}\right)^2}\right) \tag{5.16}$$

$$DistMinSB = \min\left(\sqrt{\left(X_{max} - X_{SB}\right)^2 + \left(Y_{max} - Y_{SB}\right)^2}\right) \tag{5.17}$$

Pour calculer le voisinage du nœud, nous considérons ce que nous appelons le rayon, calculé à l'aide de l'équation suivante.

$$R = \sqrt{\frac{A}{\pi \times k}} \tag{5.18}$$

Où $A = M \times M$ est l'aire du réseau et k est le nombre optimal de groupes, il peut être trouvé comme suit.

Tous les nœuds sont situés dans la zone S avec $(d < d_0)$. Par conséquent, la dissipation d'énergie pour transmettre un message l-bit dans le chef est donnée par :

$$
\begin{aligned}
E_{Chef} &= E_{Tx} + E_{Rx} + E_{DA} \\
&= \left(l \times E_{elec} + l \times \epsilon_{fs} \times d_{SB}^2 \right) + \left(\left(\frac{N}{k} - 1 \right) \times l \times E_{elec} \right) \\
&+ \left(\frac{N}{k} \times E_{DA} \right)
\end{aligned}
\tag{5.19}
$$

Où d_{SB} est la distance entre le nœud chef et la SB. L'énergie dissipée des autres nœuds est calculée comme suit :

$$
E_{nonChef} = T_{Tx} = l \times E_{elec} + l \times \epsilon_{fs} \times d_{Chef}^2 \tag{5.20}
$$

La zone de détection est formulée par plusieurs groupes qui ont les chefs comme centre. Elle est calculée comme suit :

$$
S = \left(2 \times \pi \times d_{Chef}^2 \right) \times k \simeq A \tag{5.21}
$$

De l'équation 5.21, on déduit la distance séparant un nœud du chef :

$$
d_{Chef}^2 = \frac{A}{2 \times \pi \times k} \tag{5.22}
$$

Donc :

$$
E_{nonChef} = l \times E_{elec} + l \times \epsilon_{fs} \times \frac{A}{2 \times \pi \times k} \tag{5.23}
$$

Maintenant, l'énergie dissipée dans un groupe dans une période est donnée par :

$$
E_{Groupe} = l \times \left(E_{Chef} + \frac{N}{k} \times E_{nonChef} \right) \tag{5.24}
$$

L'énergie totale est :

$$
E_{Total} = k \times E_{Groupe} = k \times l \times \left(E_{Chef} + \frac{N}{k} \times E_{nonChef} \right) \tag{5.25}
$$

Pour trouver k, le nombre de groupes formées dans un cycle, nous assimilons l'équation 5.25 à zéro et différencions par rapport à k :

$$
\begin{aligned}
k &= \left(\frac{\sqrt{N \times A}}{\sqrt{2 \times \pi}} \times \frac{1}{d_{SB}} \right) \times 100 \\
&= \frac{\sqrt{(1 - (m_1 + m_2)) \times N + m_1 \times N + m_2 \times N \times A}}{\sqrt{2 \times \pi}} \times \frac{100}{d_{SB}}
\end{aligned}
\tag{5.26}
$$

Où N est le nombre de nœuds de capteur calculé à partir de l'équation 5.39 et d_{SB} est la distance moyenne d'un nœud à la SB est donnée par [78] :

$$
d_{SB} = 0.765 \times \frac{A}{2} \tag{5.27}
$$

Selon [79], un nombre croissant de groupes conduit à une distribution de groupes de petite taille, ce qui est meilleur en termes de consommation d'énergie. Le nombre de groupes fixe augmente la stabilité d'un RCSF.

Puisque notre objectif est de maximiser la fonction de coût, pour calculer la capacité de mémoire d'un nœud, nous utilisons la fraction entre la taille initiale du tampon (Initial buffer size (IBS)) et l'espace restant du tampon (Current buffer size (CBS)).

$$Capacité(i) = \frac{IBS(i)}{IBS(i) - CBS(i)} \tag{5.28}$$

L'énergie du nœud est calculée en utilisant la fraction entre l'énergie initiale du nœud (Initial battery power (IBP)) et l'énergie restante comme dans l'équation 5.29.

$$Energie(i) = \frac{IBP(i)}{IBP(i) - CBP(i)} \tag{5.29}$$

Le dernier paramètre de la fonction de coût est calculé comme dans la formule 5.30.

$$Pertinence(i) = \frac{CBP(i)}{Energie(i) \times Localisation(i)} \tag{5.30}$$

5.2.2 La sélection des chefs basée sur WOA

Dans cette partie, la sélection du chef dans le réseau hiérarchique hétérogène a été inspirée par la structure des baleines. Le tableau 5.2 présente la similitude et la comparaison entre les baleines et le RCSF.

Table 5.2 – Similitude entre WOA et le RCSF

WOA	**RCSF**
Baleines	Nœuds de capteur
Emplacement des aliments (proies)	Fonction objective
La position des proies dans la région n'est pas connue par les baleines.	Les nœuds ne connaissent pas à l'avance la fonction de coût. Les nœuds ne peuvent détecter que les modifications des valeurs de cette fonction.
Meilleure valeur de la fonction de coût	Nœud chef.
Positions des baleines	Positions des nœuds (pos(t)=x(t),y(t))

Dans cette sous-section, nous expliquerons comment sélectionner les chefs à l'aide de WOA en utilisant l'algorithme 5.2.

5.2.2.1 Attaque des proies (Exploitation)

Le comportement d'attaque est modélisé en fonction de la stratégie d'attaque du filet à bulles, un mécanisme d'encerclement et un mécanisme de mise à jour de position en spirale. Les baleines à bosse appliquent ces deux mécanismes avec une probabilité de 50 $\%$ pour chacun. Une variable aléatoire p est introduite et varie entre $[0; 1]$. Le modèle de mise à jour est donné par l'équation 5.31.

Algorithme 5.2 : *Élection des chefs en utilisant WOA*

Entrées : Réseau de N nœuds
Sorties : Les nœuds $Chefs$
- Initialisez la population des baleines $X_i(i = 1, 2, ..., n)$
- Calculez le $coût$ de chaque agent
- Choisir le meilleur agent de recherche X_p

Tant que *itération* $<$ *nombre max d'itérations* **Faire**
  Pour chaque agent de recherche **Faire**
    - Mettre à jour a, A, C, l et p
    - Mettre à jour la position de l'agent de recherche actuel
    Si cycle = o **Alors**
      Si $p < 0.5$ **Alors**
        Si $|A| < 1$ **Alors**
          - Utilisez $X(t+1) = X_p(t) - A \times D$
        Sinon
          - Sélectionnez un agent de recherche aléatoire X_{rand}
          - Utilisez $X(t+1) = X_{rand} - A \times D$
        Fin Si
      Sinon
        - Utilisez $X(t+1) = D \times e^{bl} \times \cos(2\pi l) + X_p(t)$
      Fin Si
    Sinon
      - Utilisez les cas 1, 2, 3 ou 4
    Fin Si
  Fin Pour
  - Vérifiez si un agent de recherche dépasse l'espace de recherche et modifiez-le
  - Calculez le coût de chaque agent de recherche
  - Mettre à jour X_p s'il y a une meilleure solution
  - itération = itération + 1

Fin Tant que
Retournez X_p (Le chef est le nœud le plus proche de la position de X_p)

$$X(t+1) = \begin{cases} X_p(t) - A \times D & si \;\; p < 0.5 \\ D \times e^{bl} \times \cos(2\pi l) + X_p(t) & si \;\; p \geq 0.5 \end{cases} \tag{5.31}$$

Où D indique la meilleure solution jusqu'à présent (la distance de la i^{ème} baleine jusqu'à la proie), l est un nombre aléatoire dans la plage $[-1; 1]$, b est une valeur constante qui définit la spirale logarithmique [1] ($b = 0.1759 = \frac{\ln \epsilon}{\pi}$ pour Nautilus).

5.2.3 L'adhésion des membres aux groupes selon ICA

Le nœud considère la SB comme son prochain saut si cette dernière se trouve dans son rayon de communication, sinon il choisira le chef pour lui transmettre ses données en utilisant ICA. Pour créer des empires

1. Une spirale logarithmique est une sorte de spirale vue dans le monde naturel. Plusieurs exemples se trouvent dans les coquilles de certains mollusques, comme celle des toiles d'araignées et des ammonites fossiles ainsi que dans les nautiles.

initiaux, ICA génère un ensemble de solutions aléatoires dans l'espace de recherche du problème d'optimisation. Cette population initiale est appelée pays. Pour déterminer la puissance de chaque pays, une fonction de coût est utilisée. Sur la base de la valeur de cette fonction, les meilleurs pays initiaux deviennent impérialistes et prennent le contrôle des autres pays appelés colonies. Les impérialistes et les colonies forment ce que nous appelons un empire. L'assimilation et la révolution sont les deux principaux opérateurs de cet algorithme. L'assimilation est utilisée pour approcher l'État impérialiste. La révolution change la position de certains pays au hasard pour explorer davantage l'espace de recherche. Une meilleure position pourrait être atteinte par une colonie au cours de ces deux processus. À ce stade, elle contrôlera tout l'empire et changera l'état impérialiste actuel. Pendant le processus de compétition, de puissants empires prennent possession de leurs colonies. L'algorithme 5.3 représente le fonctionnement d'adhésion selon ICA.

Le tableau 5.3 présente la similitude et la comparaison entre l'ICA et le RCSF.

Table 5.3 – Similitude entre ICA et le RCSF

ICA	**RCSF**
Pays	Nœuds de capteur
Colonies	Nœuds normaux (non-Chef)
Empire	Groupe
Assimilation	Algorithme de sélection des chefs
Puissance de chaque pays	Fonction objective
L'impérialisme (meilleur pays) prend le contrôle de tout l'empire	Le chef joue le rôle du leader par rapport à ses membres
Pendant la compétition, des empires faibles s'effondrent et des puissants prennent possession de leurs colonies.	Les chefs avec une faible valeur de la fonction de coût laisseront leurs positions à d'autres nœuds.

5.2.3.1 Génération des empires initiaux

Dans cette sous-section, nous expliquons comment les nœuds rejoignent le meilleur chef en utilisant ICA.

Le pays est représenté par les nœuds. Les meilleurs pays sont appelés impérialistes et les moins bons sont les colonies. Le coût d'un pays est trouvé en évaluant plusieurs paramètres $(p_1, p_2, p_3, ..., p_n)$ au niveau du nœud. Dans notre cas, nous avons utilisé la fonction de coût décrite dans 5.14.

$$Coût = f(pays) = f(p_1, p_2, p_3, ..., p_n) \tag{5.32}$$

Pour démarrer l'algorithme d'optimisation, nous générons la population initiale de taille N_{pop}. Nous sélectionnons ($N_{imp} = Chefs$) les pays les plus puissants pour former les empires en utilisant l'algorithme GWO 5.1, et ($N_{col} =$ *nœuds normaux*) les pays restants de la population seront les colonies qui appartiennent

Algorithme 5.3 : *Adhésion d'un membre à l'aide d'ICA*

Entrées : N Pays
Sorties : Des colonies pour chaque impérialiste
- Initialisez les impérialistes à l'aide de GWO
Tant que *itération* $< N$ **Faire**
- Déplacez les colonies vers leur impérialiste (Assimilation) en utilisant le critère de distance.
- Calculez le coût total de tous les empires
- Choisissez la colonie la plus faible de l'empire le plus faible et donnez-la à l'empire qui est le plus susceptible de la posséder (compétition impérialiste)
Si Il y a un empire sans colonies **Alors**
Si Ce pays a beaucoup de richesses **Alors**
- Élire ce pays comme dépendant
Sinon
- Le pays demande l'aide de la colonie la plus proche
Fin Si
Fin Si
- itération = itération + 1
Fin Tant que
Retournez Des colonies pour chaque impérialiste

chacune à un empire. Après cela, les colonies sont réparties entre les impérialistes en fonction de leur pouvoir de former les empires initiaux. Pour ce faire, nous définissons le coût normalisé d'un impérialiste en :

$$Coût_{Normalisé} = coût_n - \max\left(coût_i\right) \tag{5.33}$$

Où le $coût_n$ est le coût de n^{ème} impérialiste et $coût_{Normalisé}$ est son coût normalisé. Les colonies qui devraient appartenir à l'empire sont définies par la puissance normalisée calculée par :

$$Puissance_{Normalisée} = \left| \frac{Coût_{Normalisé}}{\sum_{i=1}^{N_{imp}} coût_i} \right| \tag{5.34}$$

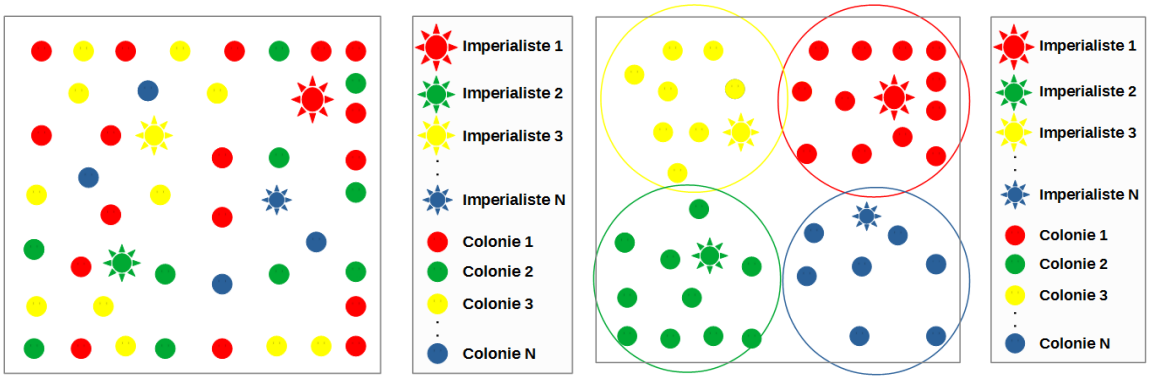
Ensuite, le nombre initial de colonies d'un empire sera :

$$InitNumCol_n = \left(Puissance_{Normalisée} \times Num_{col}\right) \tag{5.35}$$

Où Num_{col} est le nombre de toutes les colonies et $InitNumCol_n$ est le nombre initial de colonies du n^{ème} empire. Pour chaque impérialiste, nous choisissons au hasard $InitNumCol_n$ et nous les leur donnons afin de diviser les colonies. Les empires les plus faibles ont un plus petit nombre de colonies tandis que les plus grands en ont plus.

En suivant exactement cette méthode, la distribution des empires sera déséquilibrée comme le montre la figure 5.2. Nous notons aussi qu'une colonie peut appartenir à un impérialiste lointain, ce qui génère de grands moyens pour acquérir les richesses du pays colonisé. Pour remédier à ces problèmes déjà évoqués, nous ajoutons le critère de distance lorsque l'impérialiste colonise les pays. Cela signifie qu'ils choisissent uniquement un pays dans une zone spécifiée.

Nous notons également qu'il y aura certains pays qui n'appartiendront à aucun impérialiste (Empires sans colonies). Dans l'algorithme d'origine, ces empires seront éliminés. Dans notre solution, ces pays s'élisent en tant que pays dépendants qui doivent satisfaire seuls leurs besoins, s'ils ont beaucoup de richesses. Sinon, ils demandent de l'aide à la colonie la plus proche.



(a) Sans le paramètre de distance (b) Avec le paramètre de distance

Figure 5.2 – Adhésion aux groupes à l'aide d'ICA

5.2.4 Transmission de données

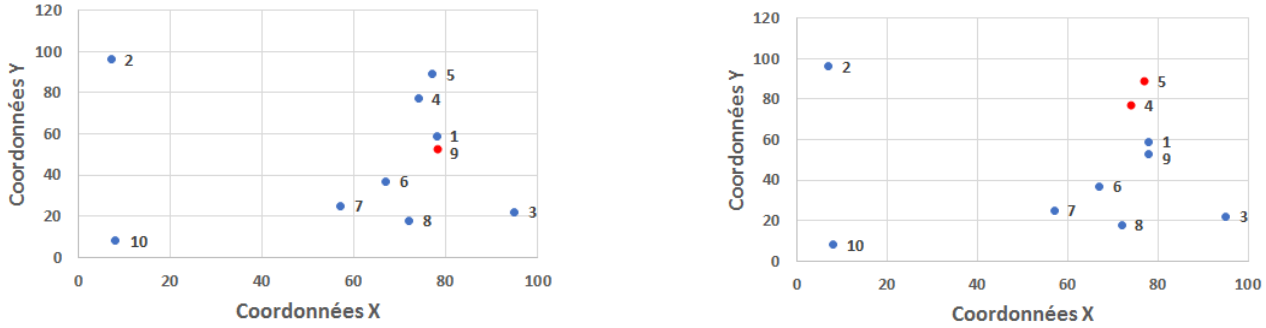
Après la phase de regroupement, les nœuds envoient leurs données au chef concerné comme dans le protocole Low energy-efficient clustering and routing based on genetic algorithm (LER-GA) [80, 81]. Le groupe, dans ce cas, est subdivisé en régions et zones. Chaque membre sait, qui sont ses prédécesseurs pour attendre de recevoir leurs données, et qui sont ses successeurs pour transmettre leurs données au niveau supérieur. Une fois que le chef a reçu toutes les données de ses membres, il les envoie à la SB.

5.3 Exemple

Nous supposons que les coordonnées de notre SB sont (500 ; 500). Le réseau est composé de 10 nœuds dispersés de manière aléatoire avec une puissance de batterie initiale de 0,75 J et une taille de tampon de 256 bits. Les nœuds seront regroupés selon GWO et WOA pour un cycle et une itération. A un instant donné, les valeurs des différents paramètres ont été enregistrées dans le tableau 5.4.
Lorsque nous appliquons l'algorithme original de GWO ou celui modifié, le chef choisi est le nœud 9. Nous remarquons sur la figure 5.3a que le chef est positionné dans un endroit stratégique pour recevoir le maximum de données. En revanche, lorsque nous appliquons l'algorithme WOA, le choix dépend de la valeur de p et A. Si $p < 0,5$ et $A < 1$, le nœud choisi comme chef est 4. Sinon, si $p \geq 0,5$, le chef est le nœud 5. La figure 5.3b représente le regroupement utilisant WOA.

Table 5.4 – Valeurs des différents paramètres dans l'exemple à t = 5ms

Nœuds	X	Y	Mémoire occupée	CBP	Coût
1	78	59	32	0,266297	113,293041842
2	7	96	16	0,265841	120,391234588
3	95	22	32	0,266285	113,142636991
4	74	77	8	0,26583	24,870323651
5	77	89	1	0,266274	64,851620409
6	67	37	64	0,265818	97,37376436
7	57	25	32	0,266263	113,473284611
8	72	18	16	0,265816	121,23507957
9	78	53	16	0,266252	121,304446822
10	8	8	16	0,265806	120,251344027



(a) Sélection des chefs avec GWO

(b) Sélection des chefs à l'aide de WOA

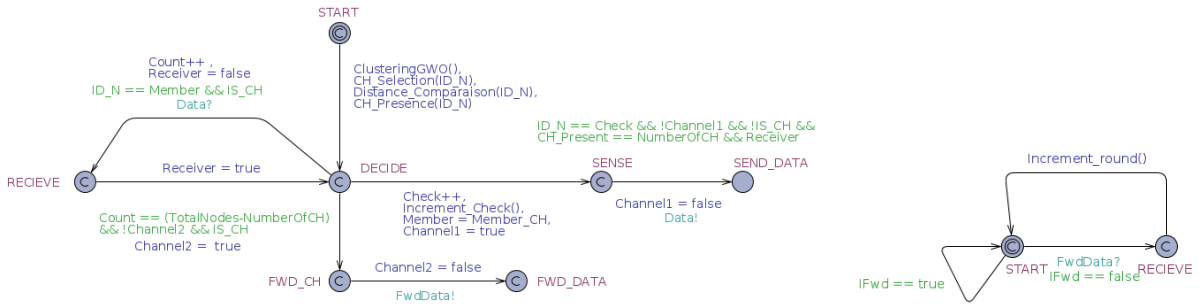
Figure 5.3 – Exemple illustratif

5.4 Modélisation et vérification

Les méthodes formelles sont utilisées pour modéliser et vérifier les systèmes complexes à l'aide d'entités mathématiques. Il existe quatre catégories de méthodes formelles : spécification formelle, preuves formelles, abstraction et vérification de modèle [82]. Cette dernière catégorie est la plus utilisée par les chercheurs, en raison de sa capacité à détecter les pires scénarios. Ces scénarios ne sont pas possibles dans les simulations informatiques et autres techniques de test. La plupart des erreurs et bugs cachés dans différents systèmes, codes et protocoles peuvent être facilement trouvés. Certaines propriétés importantes telles que l'absence d'impasse, la sûreté, la vivacité etc., doivent être garanties dans les systèmes critiques tels que les applications de E-santé, où des vies humaines sont concernées. La défaillance de certaines parties du système peut entraîner de graves dommages à l'équipement ou à l'environnement. Il est nécessaire de trouver ces bugs en utilisant la vérification formelle avant l'implémentation du système. Pour effectuer cette vérification, PRISM, SPIN et UPPAAL sont les outils les plus fréquemment utilisés [83]. Les auteurs de cette référence [83] ont recommandé d'utiliser UPPAAL dans la modélisation des protocoles de communication qui implique le temps. Nous découvrons que la diffusion en UPPAAL le rend un outil adéquat pour modéliser les protocoles des RCSFs, qui est une fonctionnalité absente dans SPIN. Alors que SPIN peut fournir des messages tamponnés et des passages de rendez-vous, une fonctionnalité qui n'est pas possible dans UPPAAL. Les

collisions et la perte de messages peuvent être modélisées dans SPIN facilement. Selon les auteurs de la référence [82], UPPAAL est plus flexible en termes de convivialité et de facilité. D'un autre côté, PRISM offre de bonnes fonctionnalités pour les modèles probabilistes.

Pour vérifier et valider notre système, nous avons modélisé les protocoles GWO / WOA pour un seul cycle, de manière simplifiée en utilisant la version UPPAAL 4.0. La figure 5.4 montre l'automatisation de la transition pour le système en un seul cycle.



(a) Nœud (b) SB

Figure 5.4 – La transition d'état des protocoles GWO / WOA pour un cycle

Le modèle de GWO / WOA a été utilisé pour prouver les propriétés essentielles décrites dans la figure 5.5. Ces preuves sont valables pour plusieurs topologies. Dans ce qui suit, nous détaillons ces propriétés vérifiées pour un réseau composé de 10 nœuds où seuls 2 chefs sont élus. Le diagramme de séquence des messages est illustré dans la figure 5.6.

Lemme 1. *Si l'automate ne boucle pas ou ne bloque pas, la propriété de non blocage est satisfaite.*

Preuve 1. Selon la figure 5.6, la phase de regroupement et la phase de transmission des données se terminent sans interruption.

Lemme 2. *Si un nœud appartient à un seul groupe et qu'un groupe n'a qu'un seul chef, la propriété de sûreté est satisfaite.*
$E <> \left(Node0.Member_{Chef} == 1 \text{ et } Node0.Member_{Chef} == 2 \right)$ (vrai si non satisfaite)

Preuve 2. L'élection du chef dépend de la valeur de la fonction de coût. Si un nœud a la valeur maximale, il sera choisi comme chef. De plus, deux nœuds ne peuvent jamais avoir une valeur de coût égale car cela dépend dans certains cas de la position des nœuds et deux nœuds ne peuvent pas être au même endroit en même temps.

Lemme 3. *Si la phase de regroupement se termine et qu'un nœud peut être un nœud ordinaire ou un chef à un instant donné, la propriété de vivacité est satisfaite.*

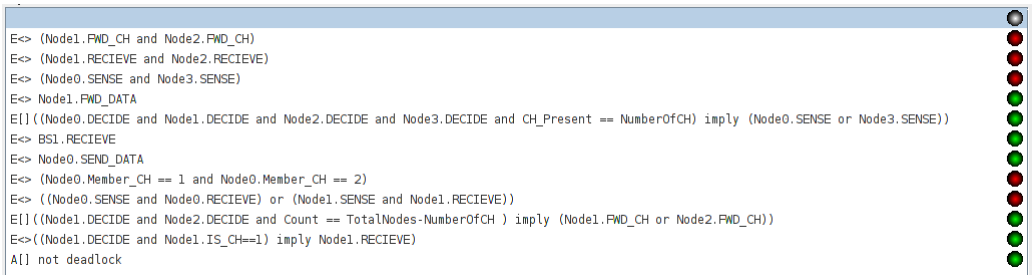


Figure 5.5 – Propriétés essentielles vérifiées sur UPPAAL

$E <> \Big((\textit{Node0.SENSE et Node0.RECIEVE}\ ) \textit{ ou } (\textit{Node1.SENSE et Node1.RECIEVE}\ ) \Big)$ (vrai si non satisfaite)

Preuve 3. Selon la figure 5.4, un processus dans l'état DECIDE ne peut pas transiter simultanément vers les états SENSE ou RECIEVE. Sa transition dépend de la garde IS-CH mise à jour dans la phase du regroupement qui se termine correctement dans lemme 1 et lemme 2.

5.5 Modèles

5.5.1 Modèle énergétique

Le modèle énergétique utilisé est comme celui utilisé dans la référence [84], où l'énergie consommée pour envoyer un message de l bits sur une distance d est calculée comme suit :

$$E_{T_x}(l,d) = \begin{cases} l \times E_{elec} + l \times \epsilon_{fs} \times d^2 & si \quad d < d_0 \\ l \times E_{elec} + l \times \epsilon_{mp} \times d^4 & si \quad d > d_0 \end{cases} \tag{5.36}$$

Où E_{T_x} est l'énergie transmise, E_{elec} est l'énergie dissipée par bit dans le circuit émetteur ou récepteur. ϵ_{fs} et ϵ_{mp} dépendent du modèle d'amplificateur de l'émetteur. Si la distance entre l'émetteur et le récepteur est inférieure à un seuil d_0, le modèle d'espace libre est utilisé ; sinon, le modèle multi-chemins est utilisé. d_0 est généralement calculé comme suit :

$$d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \tag{5.37}$$

D'un autre côté, la consommation d'énergie pour la réception d'un paquet long de l bits est calculée comme suit :

$$E_{R_x} = l \times E_{elec} \tag{5.38}$$

5.5.2 Modèle de réseau

Dans cette solution, plusieurs scénarios de RCSFs sont utilisés. Différent nœuds capteurs hétérogènes sont distribués dans la zone d'intérêt. Il existe des nœuds m_1 équipés de α fois plus de taille de tampon agissant

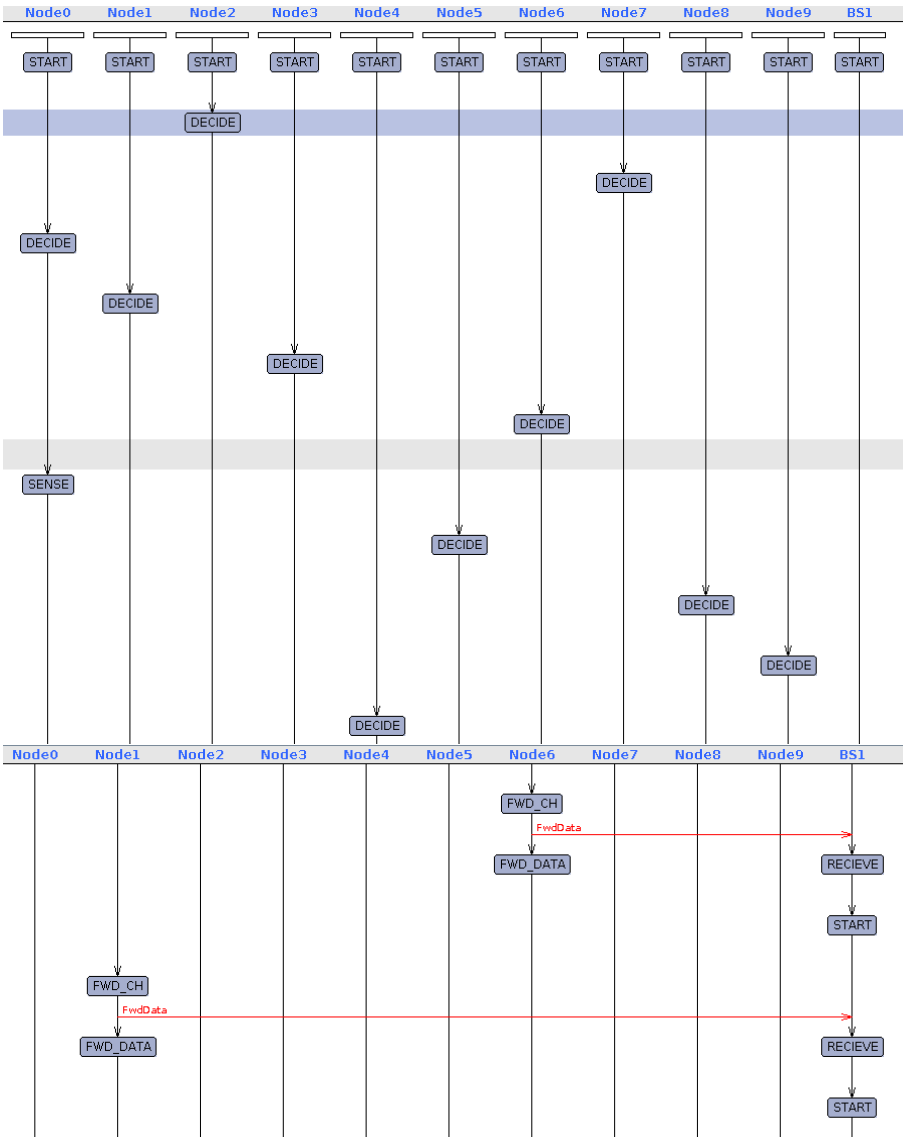


Figure 5.6 – Diagramme de séquence de messages illustrant un seul cycle sur les protocoles GWO / WOA

comme des nœuds avancés et des nœuds m_2 équipés de β fois plus d'énergie agissant comme des super nœuds. Le nombre total de nœuds (N) peut être calculé comme dans la référence [85] :

$$N = \Big(1 - \big(m_1 + m_2\big)\Big) \times N + m_1 \times N + m_2 \times N \tag{5.39}$$

Dans le réseau, les nœuds sont sélectionnés parmi les nœuds avancés selon les algorithmes de sélection des chefs basés sur GWO et WOA proposés. Les nœuds restants du réseau ont une propriété de nœud normale. Ici, il est important de noter que, dans les algorithmes proposés, chaque chef est un nœud avancé, mais tous les nœuds avancés ne peuvent pas se voir attribuer le statut de chef.

5.5.3 Modèle de système et hypothèses

Le modèle de réseau est considéré comme un modèle d'espace libre. Il a un émetteur et un récepteur avec une distance de séparation d. Les circuits d'amplification sont également présents à la fois à T_x et R_x. Le scénario du RCSF considéré pour la simulation possède toutes les propriétés et limitations suivantes pour former le modèle système :

- Tous les nœuds capteurs sont distribués de manière homogène en utilisant une distribution de loi uniforme.
- Tous les nœuds sont hétérogènes et ont une taille de tampon limitée.
- La SB est stationnaire et elle peut être situé à l'intérieur ou à l'extérieur de la zone d'intérêt.
- La fusion de données est utilisée pour minimiser le nombre de paquets de données transférées.
- Tous les nœuds déployés sont statiques, ce qui signifie qu'aucun nœud n'est capable de changer son propre emplacement une fois le processus de distribution terminé.
- Chaque nœud a une portée de communication fixe R_{Tx}.
- Le nœud avec l'énergie maximale, le nombre maximal de voisins, la taille maximale du tampon et qui se situe autour du centre de gravité du groupe est le candidat le plus approprié pour le rôle du chef.
- Les autres nœuds ne satisfaisant pas ces critères ont une faible probabilité ou aucune chance de devenir chef.
- Si deux nœuds ont les mêmes chances et sont dans le voisinage, l'un d'eux s'éteindra à ce stade pour aider à prolonger la durée de vie du réseau.
- Si la distance entre un nœud et son chef correspondant est supérieure à sa distance à la SB, ce nœud enverra ses informations détectées directement à la SB.
- Il faut noter qu'à mesure que le nombre de paramètres augmente, par conséquent, les calculs, le temps et la consommation d'énergie augmentent.

5.6 Évaluation des performances de GWO et WOA

Les simulations initiales sont faites sur le RCSF #1 avec 100 nœuds. Ensuite, elles ont été effectuées sur le RCSF #2 avec 50 nœuds et le troisième scénario sur le RCSF #3 avec 250 nœuds. La position de la SB

a également été variée pour différents scénarios. Dans le premier, elle était positionnée au centre du réseau à $(50; 50)$, puis dans le deuxième scénario, elle était positionnée à la frontière de la zone de couverture $(100; 100)$ et enfin, dans le troisième scénario, elle était positionnée en dehors du réseau à $(50; 150)$. Les différents paramètres pris en compte pour les simulations sont donnés dans le tableau 5.5.

Table 5.5 – Paramètres de simulation

Paramètres	**Valeurs**
Zone ciblée (m^2)	100 × 100
Position de la SB	(50,50), (100,100), (50,150)
Nombre de nœuds	50 - 100 - 250
Énergie initiale du nœud (J)	2 ou 4
Taille du tampon (bits)	2^{24}
Émetteur/Récepteur électroniques - E_{elec} (nJ/bit)	50
Amplificateur émetteur (espace libre) - ϵ_{fs} ($pj/bit/m^2$)	100
Amplificateur émetteur (multi-trajets) - ϵ_{mp} ($pJ/bit/m^4$)	0.013
Coût énergétique d'agrégation des données - E_{DA} (nJ/bit)	50
Taille du paquet (bits)	4000
Nombre d'itérations	Variable

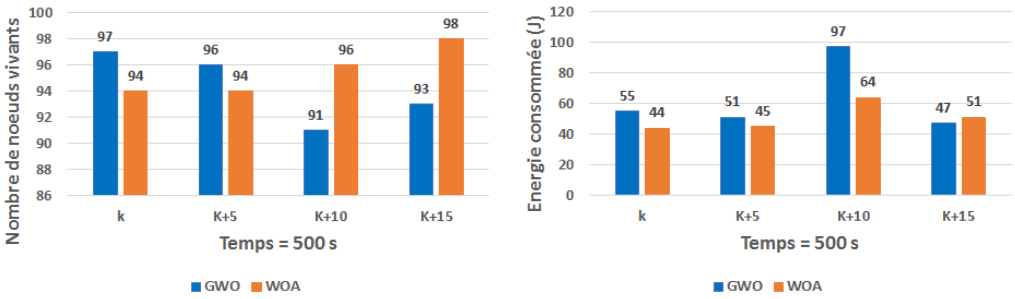
L'algorithme a été exécuté 20 fois et la moyenne des instances des données résultantes a été choisie pour tracer les résultats. Les simulations ont été faites sur la base de nombreux paramètres et en faisant varier plusieurs conditions. Premièrement, nous avons modifié les paramètres des protocoles GWO et WOA. Ensuite, nous avons comparé ces deux protocoles ensemble pour étudier l'énergie consommée et le nombre de paquets reçus par la SB. Enfin, ces deux algorithmes ont été comparés à Fuzzy-GWO [56] de la littérature. Les résultats et l'analyse de ces simulations sont présentés dans les sous-sections suivantes.

5.6.1 Variation du nombre d'itérations

La figure 5.7 représente les résultats obtenus lors du changement du nombre d'itérations. Notez qu'au temps $t = 100s$, le nombre de nœuds vivants est plus grand lorsque le nombre d'itérations est égal au nombre de groupes k pour l'algorithme GWO. L'opposé de cette observation est noté lorsque l'algorithme utilisé est WOA. Idem pour la quantité de données reçues par la SB qui est également phénoménale lors de l'utilisation de GWO et très faible lors de l'utilisation de WOA. Lorsque le nombre d'itérations est égal à $k+5$, $k+10$ et $k+15$, le nombre de nœuds vivants et le nombre de paquets reçus diminuent séquentiellement pour l'algorithme GWO et augmentent pour l'algorithme WOA. Cette différence de valeurs entre les deux algorithmes peut s'expliquer par le fait que l'algorithme GWO utilise moins de paramètres que l'algorithme WOA. Ce qui lui permet d'explorer plus rapidement l'espace de recherche et par conséquent réduire la complexité de calcul. D'un autre côté, l'algorithme WOA a besoin de plus d'itérations pour arriver à des résultats satisfaisants en termes de durée de vie et de quantité de données.

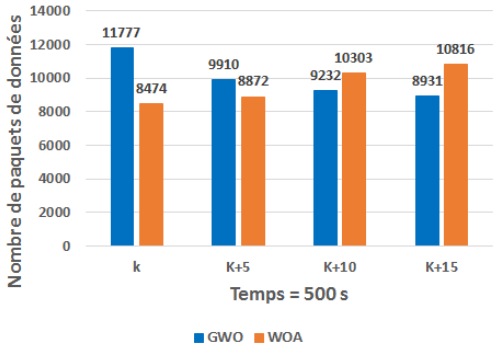
5.6.2 Complexité temporelle

La figure 5.8 représente la complexité temporelle estimée pour trouver les chefs dans un réseau homogène et hétérogène, avec deux cycles différents, en utilisant les deux algorithmes GWO et WOA. Selon les courbes, nous remarquons qu'au premier cycle, en utilisant un réseau homogène, l'algorithme GWO prend plus de



(a) Durée de vie

(b) Consommation énergétique



(c) Paquets de données reçus

Figure 5.7 – Nombre d'itérations de GWO et WOA

temps pour choisir les chefs que l'algorithme WOA. Cette observation est différente lorsque le réseau est hétérogène. Idem pour le cycle 100, où l'algorithme GWO prouve ses performances en temps de calcul par rapport à l'algorithme WOA. La complexité de calcul globale des algorithmes GWO / WOA développés est :

Théorème 1. *La complexité temporelle des algorithmes GWO et WOA est* $O\Big(k \times \big(n^2 + n \times \log n\big)\Big)$ *où* n *est le nombre de loups gris / baleines et* k *est le nombre maximum d'itérations.*

Preuve. Le mécanisme de tri, le nombre d'itérations et le nombre de loups gris / baleines, déterminent la complexité de calcul des algorithmes GWO / WOA développés. Nos algorithmes implémentent le mécanisme de tri rapide. La complexité pour le meilleur cas est $O(n \log n)$ et pour le pire cas est $O(n^2)$.

En raison d'une meilleure convergence, d'une exploration élevée, d'une faible complexité de calcul, d'un évitement des minima locaux plus rapide et d'un temps de calcul réduit requis pour résoudre des problèmes réels, les algorithmes GWO / WOA proposés peuvent être utilisés pour résoudre des problèmes d'optimisation à grande échelle. De plus, à partir de la figure 5.7, nous avons $k_{loup} < k_{baleine}$. Par conséquent, $O(GWO) < O(WOA)$. L'algorithme GWO est nettement meilleur en termes de temps de calcul, ce qui le favorise, en particulier pour les réseaux de topologie hétérogène.

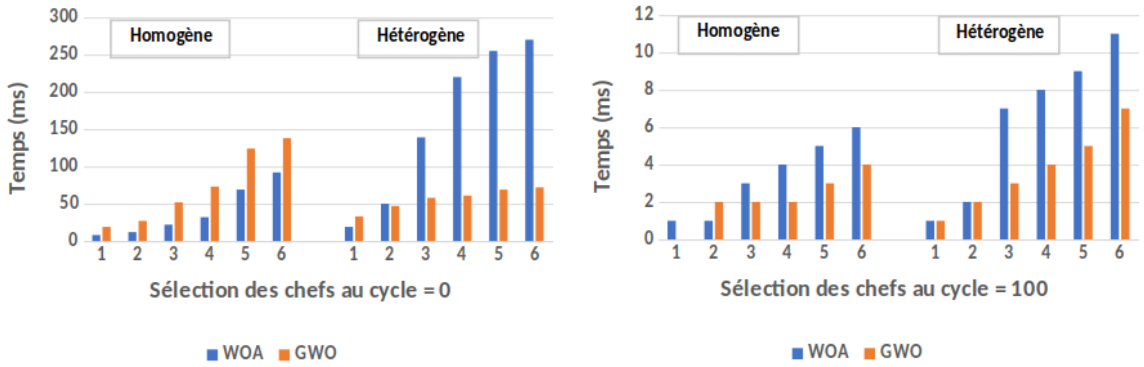


Figure 5.8 – Complexité temporelle de GWO et WOA

5.6.3 Variation du paramètre *a*

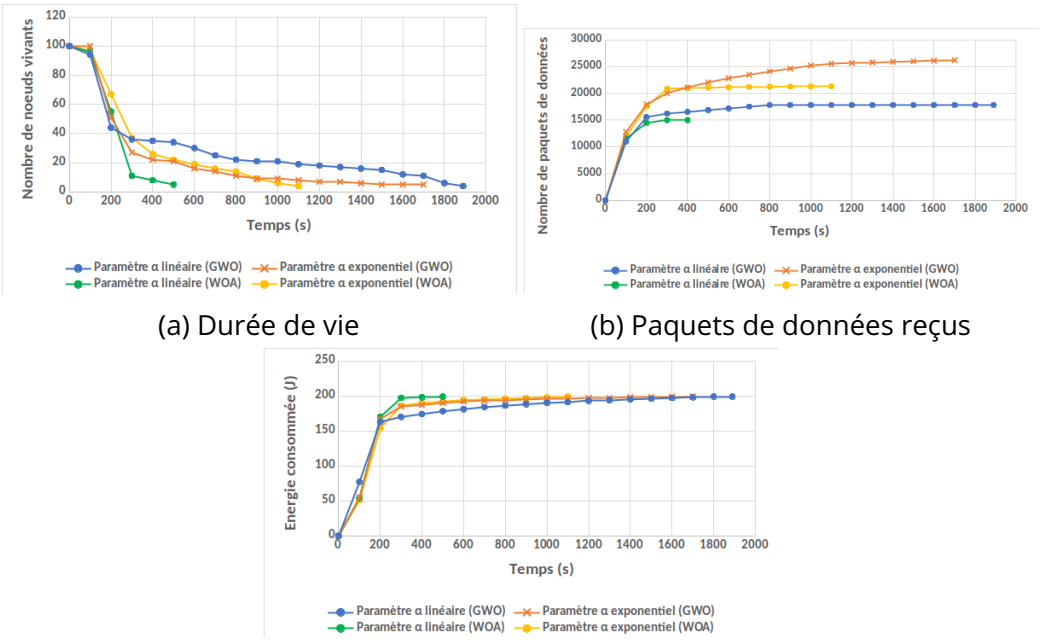
La figure 5.9 représente les résultats obtenus lorsque le paramètre a est modifié à l'aide des deux formules 5.5 et 5.6. Nous constatons sur cette figure que, lorsque le temps est inférieur à 200 s, le nombre de nœuds vivants décroît rapidement. En parallèle, le nombre de paquets reçus par la SB et l'énergie consommée augmentent rapidement. Ceci est observé pour les deux valeurs de a en utilisant les deux algorithmes. Cette consommation d'énergie croissante est due au nombre de paquets envoyés à la SB, une moyenne de 17000 paquets envoyés en seulement 200 s. Au-delà de ce temps, les nœuds meurent progressivement pour les deux valeurs de a. Cela se produit lorsque la valeur de a est modifiée dans l'algorithme GWO. Le contraire de cette observation est observé lors de l'utilisation de l'algorithme WOA. En ce qui concerne la consommation d'énergie, elle change presque de manière similaire pour les deux valeurs de a dans les deux algorithmes. La quantité de données est beaucoup plus importante de $\frac{1}{3}$ par rapport à celle obtenue lorsque a est changé pour les deux algorithmes. Contrairement aux résultats obtenus par l'algorithme GWO, lorsque nous modifions la valeur de a par la formule 5.6 dans l'algorithme WOA, la durée de vie augmente et la quantité de données est considérable. Cela peut s'expliquer par le fait que lors de l'utilisation de la formule 5.6, nous diminuons la valeur de a de façon exponentielle au lieu de le faire de façon linéaire comme dans la formule 5.5, ce qui nous permet d'explorer l'espace de recherche beaucoup plus rapidement.

5.6.4 Le passage à l'échelle

La figure 5.10 représente le temps consacré au regroupement en utilisant les deux algorithmes dans plusieurs tailles de réseau (50, 100 et 250). Notez que dans un réseau de 50 nœuds, les courbes sont superposées, ce qui indique que le temps de regroupement est le même pour les deux algorithmes. Pour un réseau de 100 nœuds, l'algorithme WOA est plus avantageux dans le temps que son homologue l'algorithme GWO. En revanche, au-delà de cette taille, cette remarque est inversée. L'algorithme GWO offre plus de performances lorsque la taille du réseau augmente.

5.6.5 La fonction de coût vs le nombre d'itérations

La figure 5.11 représente la relation entre le nombre d'itérations et la valeur de coût. Nous notons que plus le nombre d'itérations augmente, plus la valeur du coût est améliorée, sauf qu'elle affecte le temps de calcul



(a) Durée de vie

(b) Paquets de données reçus

(c) Consommation énergétique

Figure 5.9 – Différentes valeurs du paramètre a pour GWO et WOA

qui va inévitablement augmenter. Par conséquent, il est nécessaire de trouver un compromis entre le nombre d'itérations et l'optimum souhaité.

5.6.6 Réseau homogène vs réseau hétérogène

La figure 5.12 représente les résultats obtenus pour les deux algorithmes GWO et WOA dans des réseaux homogènes et hétérogènes. Nous remarquons qu'il n'y a pas de grande différence entre les résultats. Les courbes sont presque superposées. Sauf que, dans un réseau hétérogène, l'algorithme GWO délivre plus de paquets par rapport à l'algorithme WOA.

5.6.7 Variation de la localisation de la SB

La figure 5.13 représente les résultats obtenus lorsque nous modifions l'emplacement de la SB (50 ; 50), (100 ; 100) et (50 ; 150). Les courbes (a), (c) et (e) sont celles obtenues par l'algorithme GWO. Les autres courbes proviennent de l'algorithme WOA. Les résultats montrent que plus la SB est éloignée de la zone surveillée, plus la durée de vie des capteurs est courte. La quantité de données est importante lorsque la SB est située au centre du réseau.

5.6.8 Variation de *x(t+1)*

La figure 5.15 représente les résultats obtenus lorsque nous modifions la valeur de $x(t+1)$ en utilisant les formules 5.10, 5.11, 5.12 et 5.13. À partir des courbes des deux algorithmes, nous remarquons qu'en

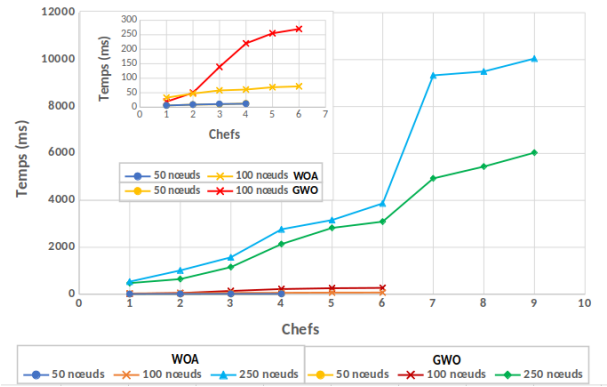


Figure 5.10 – Variation de la taille du réseau

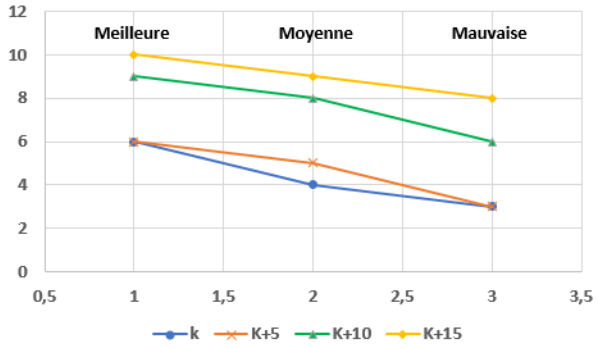
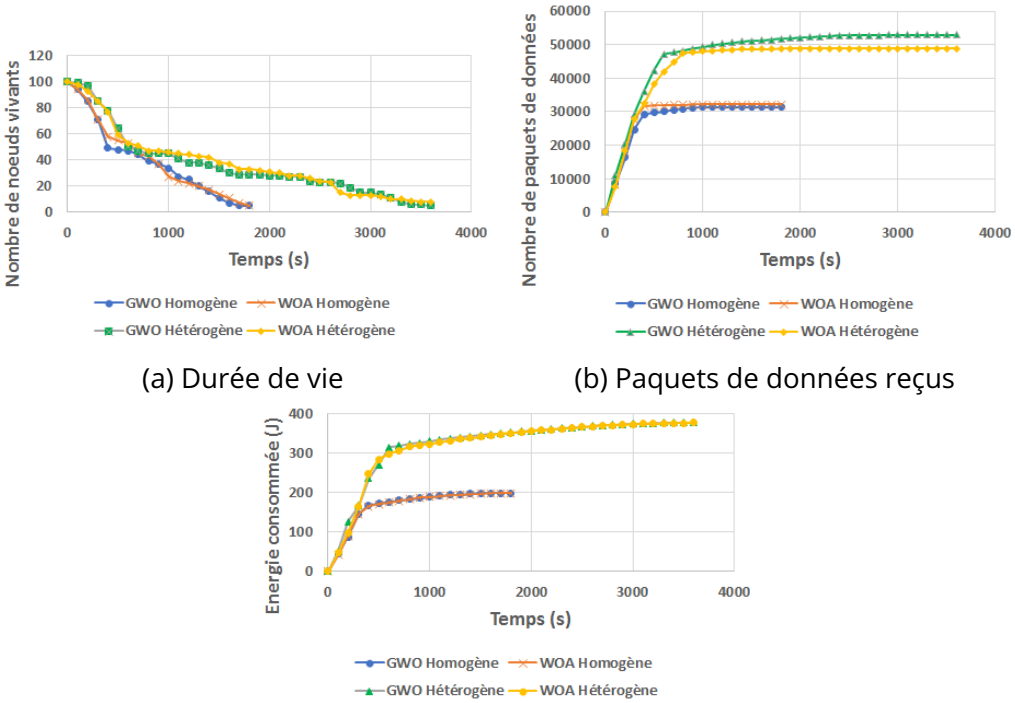


Figure 5.11 – Les valeurs de la fonction de coût sur 20 exécutions indépendantes pour k, k + 5, k + 10 et k + 15 itérations

utilisant la formule 5.10, la durée de vie atteint les 2000 s par rapport aux autres formules. Bien que la quantité de données soit nettement meilleure lors de l'utilisation des formules 5.12 et 5.13. Dans ce cas, le choix de $x(t+1)$ dépend en grande partie de l'objectif que nous voulons atteindre. Si nous privilégions le facteur énergétique, la formule 5.10 sera le bon choix. Si c'est la quantité de données reçues, les formules 5.12 et 5.13 seront les mieux adaptées. La formule 5.11 est déconseillée lors de l'utilisation de GWO.

5.6.9 GWO vs WOA vs Fuzzy-GWO

La figure 5.16 représente une comparaison en termes de durée de vie et d'acheminement de données entre les algorithmes GWO, WOA et Fuzzy-GWO [56]. Notez que dans l'algorithme Fuzzy-GWO, le premier nœud meurt à $t = 1000$ s, contrairement aux autres protocoles où la mort des nœuds est progressive. D'un autre côté, les algorithmes GWO et WOA acheminent 4 fois plus de quantité de données à la SB.



(a) Durée de vie

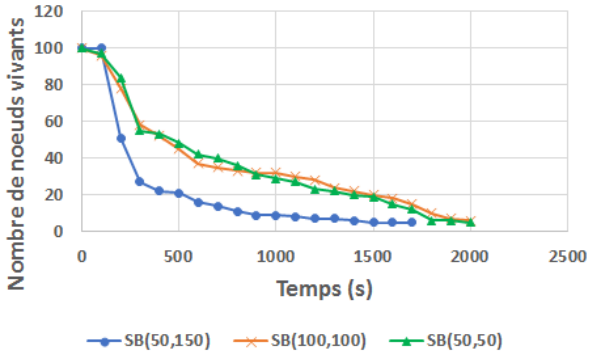
(b) Paquets de données reçus

(c) Consommation énergétique

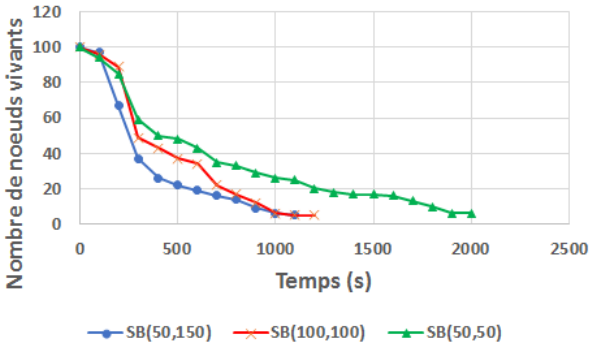
Figure 5.12 – GWO et WOA dans un réseau homogène et hétérogène

5.7 Exemple d'application

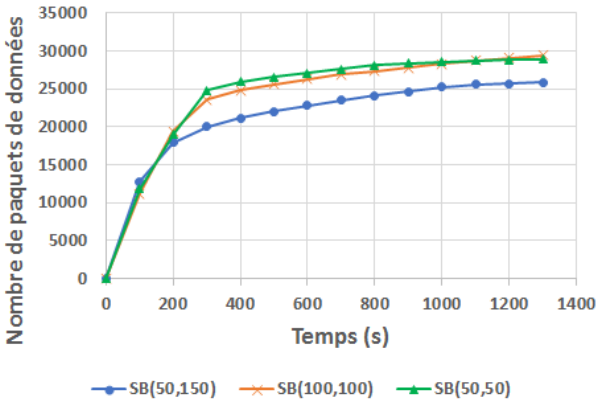
Cet algorithme peut être une bonne solution à appliquer dans le E-santé, qui utilise un type particulier de capteurs nommé Capteur corporel sans fil (CCSF). Contrairement au RCSF, ces capteurs peuvent être très limités en termes de puissance énergétique et de traitement et, par conséquent, la préservation des capacités des nœuds est d'une grande importance. De plus, afin de minimiser les interférences et de faire face à des problèmes de santé tels que le réchauffement des tissus de la peau des patients, une puissance de transmission extrêmement faible des nœuds est requise. L'énergie n'est pas le seul facteur à prendre en compte dans ce type d'application, il existe d'autres facteurs tels que la topologie, la température, la posture, la portée radio des capteurs, la QoS appropriée et la sécurité des données. Ces deux derniers points sont très complémentaires. Assurer la sécurité des informations des patients et la confidentialité des utilisateurs de données sur les réseaux sans fil nécessite l'utilisation d'algorithmes de chiffrement complexes. Le dilemme de fiabilité est particulièrement important dans ce cas. Pour être très fiable, des frais généraux élevés en termes de taille de données, de consommation d'énergie et de passage à l'échelle sont nécessaires. Notre protocole parvient à trouver un compromis entre ces trois points en même temps. Les paramètres du protocole doivent être définis pour être appliqués au CCSF. Le nombre de capteurs, qui devraient être placés dans le réseau, est supérieure à 50 et peut atteindre 256 comme défini dans IEEE 802.15.6. De plus, en raison de la courte portée de communication dans ce type de réseau, le rayon du nœud doit être fixé à 3 mètres. L'hétérogénéité des capteurs en termes d'énergie disponible, de puissance de calcul et de taille de tampon est déjà prise en compte dans notre solution.



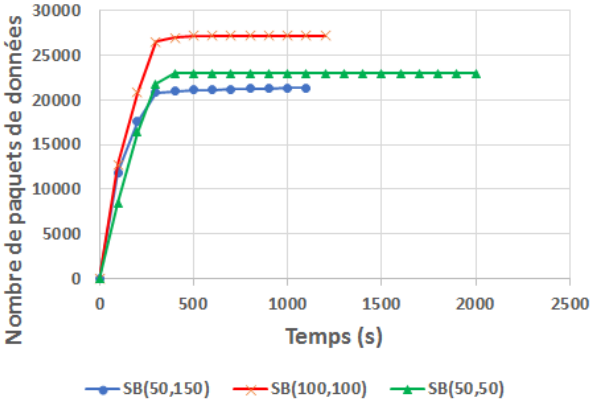
(a) Durée de vie (GWO)



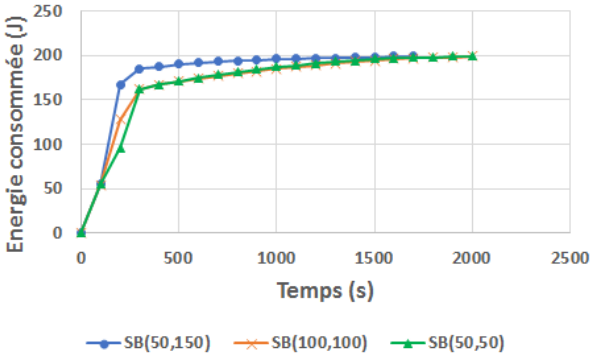
(b) Durée de vie (WOA)



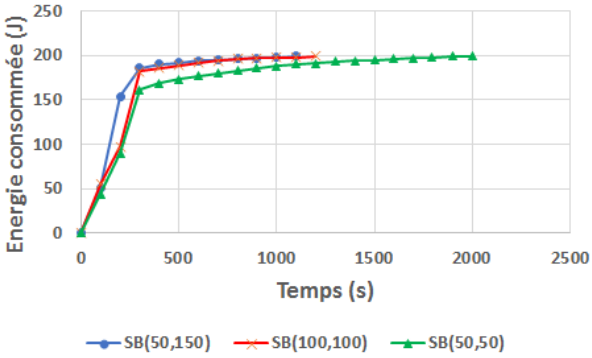
(c) Paquets de données reçus (GWO)



(d) Paquets de données reçus (WOA)

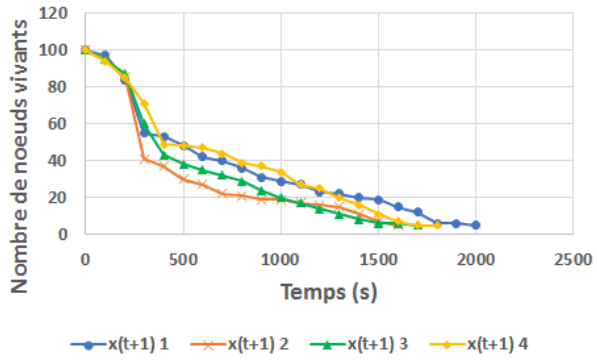


(e) Consommation énergétique (GWO)

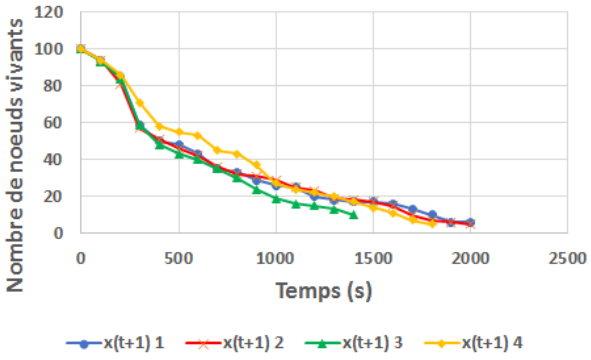


(f) Consommation énergétique (WOA)

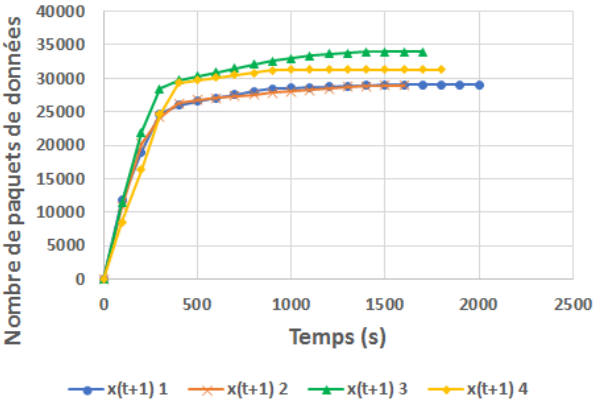
Figure 5.13 – Différents emplacements pour la SB pour GWO et WOA



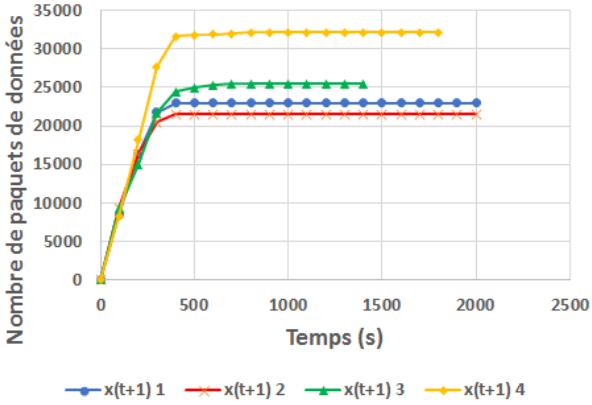
(a) Durée de vie (GWO)



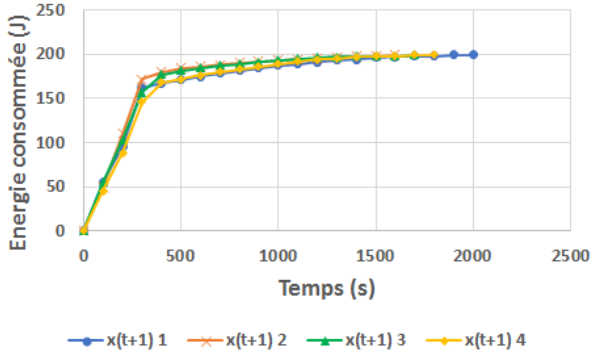
(b) Durée de vie (WOA)



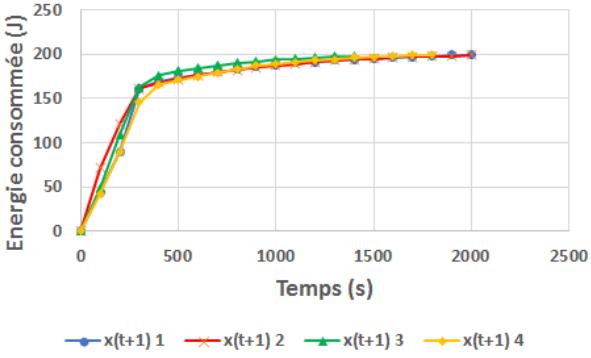
(c) Paquets de données reçus (GWO)



(d) Paquets de données reçus (WOA)

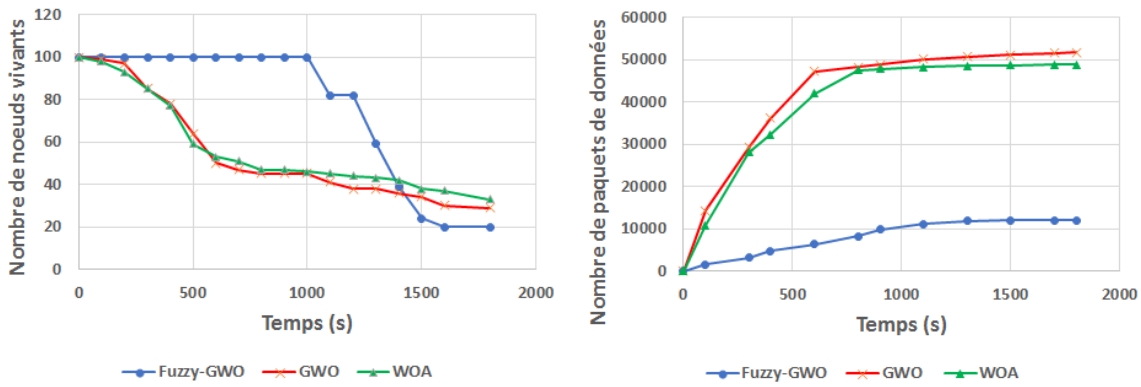


(e) Consommation énergétique (GWO)



(f) Consommation énergétique (WOA)

Figure 5.15 – Différentes valeurs de $x(t+1)$ pour GWO et WOA



(a) Durée de vie

(b) Paquets de données reçus

Figure 5.16 – Acheminement de données dans GWO vs WOA vs Fuzzy-GWO

5.8 Conclusion

Dans ce chapitre, une nouvelle approche pour la sélection des chefs dans un réseau hétérogène a été proposée pour résoudre le problème de dépassement de tampon lors de la collecte de données. Cette approche est basée sur deux algorithmes, l'algorithme d'optimisation des loups gris (GWO) et l'algorithme d'optimisation des baleines (WOA), combinés avec l'algorithme compétitif impérialiste (ICA). La bonne gestion des tampons à l'aide de ces techniques bio-inspirées a conduit à une réduction significative de la perte de données. Le protocole a été comparé à l'aide de différents paramètres et à d'autres algorithmes tels que Fuzzy-GWO.

Les résultats expérimentaux ont révélé que GWO / WOA a surpassé l'algorithme Fuzzy-GWO dans la plupart des cas. De plus, nous avons déduit que l'algorithme GWO est plus avantageux pour les réseaux à grande échelle et que l'algorithme WOA est plus efficace dans les réseaux hétérogènes ou le nombre de nœuds est inférieur à 100 nœuds.

Chapitre 6

Mouvement du collecteur de données inspiré par les salpes

Sommaire

6.1 Modified salp swarm algorithm (MSSA) 67
6.2 Évaluation des performances de MSSA 70
6.3 Conclusion 75

Dans ce chapitre, nous présentons un nouveau modèle de mobilité du collecteur de données nommé Modified salp swarm algorithm (MSSA) [86]. Ce modèle imite le mouvement des salpes pour réduire la dissipation d'énergie, la perte de données du au dépassement de tampon des nœuds chefs, et la vitesse de convergence dans un réseau hétérogène. Par conséquent, MSSA introduit un nouveau paramètre de contrôle, l'inertie α, pour ajuster la meilleure solution trouvée.

6.1 Modified salp swarm algorithm (MSSA)

En 2017, Mirjalili et al. [87] ont suggéré un nouveau protocole basé sur la population des salpes appelé Salp swarm algorithm (SSA). Cet algorithme reproduit le mécanisme d'alimentation des salpes dans les océans. Dans la mer, ces poissons forment ce qu'on appelle la chaîne des salpes. Toutes les salpes formant la chaîne sont des suiveurs sauf celle se trouvant à l'avant qui s'appelle le leader. Une interprétation de la chaîne salpe est exposée dans la figure 6.1.
Semblable à d'autres méthodes basées sur l'essaim, la position du salpe est définie dans un espace de recherche. Par conséquent, sa valeur est stockée dans une matrice à deux dimensions appelée X. Il y a aussi une source de nourriture appelée F étant cible de l'essaim.

Le salpe leader peut mettre à jour sa position en utilisant l'équation 6.1, où la signification de chaque symbole est indiquée dans le tableau 6.1.

$$X_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j) & Si \quad c_3 \geq 0 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j) & Si \quad c_3 < 0 \end{cases} \tag{6.1}$$

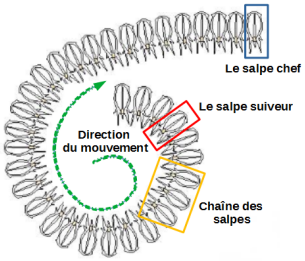


Figure 6.1 – Notion du leader et du suiveur dans la chaîne des salpes

Table 6.1 – Signification des symboles

Symboles	Significations
x_j^1	Position du leader dans la j^{ème} dimension
F_j	Position de la source de nourriture dans la j^{ème} dimension
ub_j	Borne supérieure de la j^{ème} dimension
lb_j	Borne inférieure de la j^{ème} dimension
c_2, c_3	Variables aléatoires produites uniformément dans l'intervalle [-1;1].
l	Itération actuelle
L	Nombre maximum d'itérations
x_j^i	Position du i^{ème} salpe suiveur dans la j^{ème} dimension
t	Temps
v_0	La vitesse initiale

$$c_1 = 2e^{-(\frac{4l}{L})^2} \tag{6.2}$$

Le cœfficient c_1 est le paramètre crucial dans l'algorithme SSA. Il équilibre entre l'exploitation et l'exploration. Les équations suivantes sont utilisées pour mettre à jour la position des salpes suiveurs :

$$X_j^i = \frac{1}{2}at^2 + v_0 t \tag{6.3}$$

Où $i \geq 2$; $a = \frac{v_{final}}{v_0}$ et $v = \frac{x - x_0}{t}$. Vu que dans le domaine d'optimisation, l'itération représente le temps, et considérant que $v_0 = 0$, cette équation peut être raffinée comme suit :

$$X_j^i = \frac{1}{2}(X_j^i + X_j^{i-1}) \tag{6.4}$$

Comme spécifié par Mirjalili et al. [87], la position du salpe est modifiée à l'aide des équations (6.1) et (6.4). Dans l'algorithme SSA, il existe une relation intense entre la meilleure solution actuelle et la solution mise à jour. Un nouveau paramètre α pris dans l'intervalle $[0.5; 1.5]$ est introduit dans SSA pour obtenir le SSA modifié. Ce nouveau paramètre accélère la vitesse de convergence lors de la recherche. Il établit également un équilibre entre l'exploitation et l'exploration pour échapper d'abord à un nombre énorme de solutions locales et deuxièmement pour obtenir une précision de la solution optimale.

Nous supposons que notre réseau est subdivisé en plusieurs groupes en utilisant l'algorithme GWO vu dans le chapitre précédant (chapitre 5, page 40). Notre objectif est que le collecteur de données ne collecte que

les données des chefs en se déplaçant comme un salpe. Étant donné que nous n'utilisons qu'un seul collecteur de données, qui représentera le salpe leader, nous n'avons besoin que de mettre à jour sa position. La formule modifiée est indiquée par les équations 6.5 et 6.6. Si plusieurs collecteurs de données sont utilisés dans le réseau, l'intégration des salpes suiveurs sera obligatoire dans ce cas. Chaque salpe représentera un collecteur de données mobile.

$$X_{t+1}^{1} = \begin{cases} \alpha \times F + c_1 \times c_2 \times X_t^1 & si\ c_3 \geq 0.5 \\ \alpha \times F - c_1 \times c_2 \times X_t^1 & si\ c_3 < 0.5 \end{cases} \tag{6.5}$$

$$Y_{t+1}^{1} = \begin{cases} \alpha \times F + c_1 \times c_2 \times Y_t^1 & si\ c_3 \geq 0.5 \\ \alpha \times F - c_1 \times c_2 \times Y_t^1 & si\ c_3 < 0.5 \end{cases} \tag{6.6}$$

En outre, pour améliorer la collecte de données, le collecteur se déplacera d'abord vers le chef ayant le plus grand nombre de membres. Pour ce faire, après la phase de regroupement, les chefs diffusent le message $INFO_PACKET$ contenant leurs identités et le nombre de nœuds de leur groupe. Le collecteur utilise ces informations pour mettre à jour ses coordonnées. Un exemple de ce paquet est montré par la figure 6.2.

id_chef	Degré_chef

Figure 6.2 – Paquet diffusé par le chef après la fin de la phase de regroupement

En choisissant d'aller vers le groupe ayant le plus grand nombre de membres, c'est-à-dire allez vers le chef ayant le plus petit espace mémoire restant, le problème de surcharge du tampon peut être diminué. Le paramètre F sera représenté par la valeur maximale de $Degree_{Chef}$ telle que formulée dans l'équation 6.8, où $Degree_{Chef}$ est le nombre de membres du groupe calculé par l'équation 6.7 , $Membres$ est le membre du groupe et M est le nombre de capteurs connectés à un chef.

$$Coût = Degree_{Chef} = \sum_{i=0}^{i=M} (Membres) \tag{6.7}$$

$$F = \max(Degree_{Chef}) \tag{6.8}$$

Notez que la complexité de calcul de l'algorithme MSSA 6.1 est calculée comme suit.

Théorème. *La complexité de l'algorithme MSSA est $O(n)$, où n est le nombre d'itérations.*

Preuve. Dans l'algorithme MSSA, la phase d'initialisation est égale à $O(1)$. Le calcul de la valeur du cout du collecteur mobile et de la valeur c_1 est aussi égale à $O(1)$. La mise à jour de la position du salpe leaderx_1 se réitère n fois. La complexité temporelle totale de la MSSA est donc de $O(n)$.

Algorithme 6.1 : *MSSA : Mouvement du collecteur de données*

Entrées : Réseau de N nœuds
Sorties : Mobilité du collecteur de données
- Initialiser la position du salpe leader x_1
Tant que *itération* $<$ *itérationsMax* **Faire**
- Calculez la valeur du $coût$ de chaque groupe par Eq. 6.7
- F = le meilleur groupe
- Mettre à jour le paramètre c_1 par Eq. 6.2
- Mettre à jour la position de x_1 par Eq. 6.5 et 6.6
- itération = itération + 1
Fin Tant que
Retournez Position du x_1

6.2 Évaluation des performances de MSSA

6.2.1 Hypothèses

- Les paquets perdus du au bruit du canal ne sont pas traités dans ce travail.
- Un réseau hiérarchique est utilisé.
- La durée inter-visites est supprimée.
- Le délai de livraison est réduit en augmentant l'espace tampon disponible au niveau des nœuds.

6.2.2 Initialisation du réseau

Pour évaluer les performances de l'algorithme proposé, une expérience de simulation est réalisée. Nous avons supposé que la zone d'intérêt soit de $100 \times 100m^2$ avec 100 nœuds hétérogènes déployés au hasard. Le collecteur de données est initialement positionné au centre de la zone d'intérêt. Le nœud de capteur envoie les paquets de données collectés à son chef. Après cela, ils seront redirigés vers le collecteur de données. Les différents paramètres considérés pour la simulation sont donnés dans le tableau 6.2. Dans cette expérience, la collecte de données est effectuée sur la base de l'algorithme de regroupement GWO [69]. L'algorithme MSSA est comparé à lui-même en utilisant différentes valeurs du paramètre α, un nombre différent d'itérations et de périodes de collecte de données. Ensuite, il est comparé à l'algorithme original SSA [87] pour étudier les effets et l'avantage que le paramètre α ajoute à l'algorithme. Enfin, il est comparé à l'algorithme SMBFOA [12].

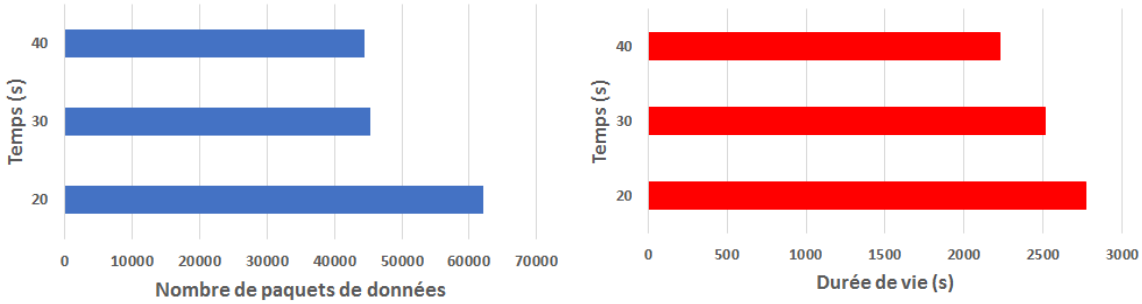
6.2.3 Variation du temps de collecte de données

La figure 6.3 représente le nombre de paquets de données collectés par le collecteur et la durée de vie du réseau en utilisant l'algorithme MSSA en changeant les moments de la collecte de données. D'après les courbes, plus la période de collecte des données est espacée, plus de paquets de données sont perdus et plus l'énergie consommée est importante. Cette observation est due au fait que les tampons du capteur sont surchargés, ce qui génère une perte de données remarquable. Dans ce cas, les nœuds sont obligés de retransmettre les paquets perdus, ce qui entraîne une consommation d'énergie accrue et une diminution

Table 6.2 – Paramètres de simulation

Paramètres	Valeurs
Zone d'intérêt (m^2)	100 × 100
Position du collecteur de données	(50,50)
Nombre de nœuds	100
Énergie initiale du nœud (J)	2 ou 4
Transmetteur/Récepteur électroniques - E_{elec} (nJ/bit)	50
Amplificateur émetteur (espace libre)- ϵ_{fs} $(pj/bit/m^2)$	100
Amplificateur d'émetteur (multi-sauts) - ϵ_{mp} $(pJ/bit/m^4)$	0.013
Coût énergétique de l'agrégation des données - E_{DA} (nJ/bit)	50
Taille du paquet (bits)	8000
Le collecteur de données récupère les données chaque (s)	20 - 30 - 40
Nombre d'itérations	20 - 50 - 100
α	$\in$ [0.5;1.5]
c_1,c_2	$\in$ [0 ; 1]

de la durée de vie du réseau. Plus la collecte est fréquente, moins la perte de paquets et d'énergie sera importante. Le meilleur moment pour collecter des données est celui effectué tous les 20 s.

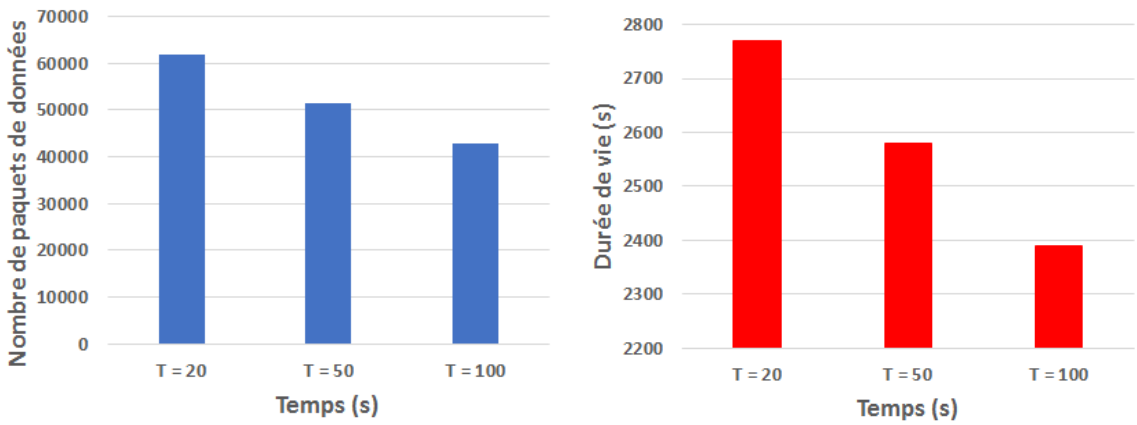


(a) Paquets de données collectés (b) Durée de vie

Figure 6.3 – Variation de la durée de collecte de données

6.2.4 Variation du nombre d'itérations

La figure 6.4 représente le nombre de paquets de données délivrés au collecteur de données ainsi que la durée de vie du réseau en variant le nombre d'itérations dans l'algorithme MSSA. Plus le nombre d'itérations augmente, plus nous nous rapprochons de la solution optimale. Cependant, cela influence négativement les performances du réseau à cause de l'augmentation de la complexité du MSSA. Nous remarquons également que le nombre de paquets et la durée de vie diminuent. De ce fait, il faut trouver un compromis entre les performances souhaitées et le nombre d'itérations.
Pour le reste des résultats de la simulation, le collecteur de données se déplace à une vitesse constante et fixe la durée de collecte de données pour tous les nœuds chefs à 20s. Le nombre d'itérations dans l'algorithme MSSA est égale à 20.



(a) Paquets de données collectés (b) Durée de vie

Figure 6.4 – Variation du nombre d'itérations

6.2.5 Variation du paramètre α

La figure 6.5 représente le premier nœud mort (First node dead (FND)), le nœud mort au milieu de la simulation (Middle node dead (MND)) et le dernier nœud mort (Last node dead (LND)) en utilisant différentes valeurs du paramètre α. Selon les courbes bleues, le premier nœud est mort à $t = 80s$ lorsque $\alpha = 1,1$. Le réseau tient jusqu'à $t = 140s$ lorsque la valeur de α est égale à $0,6$. D'après les courbes rouges, la moitié des nœuds s'éteint à $t = 1000s$ lorsque $\alpha = 0,9$, par contre, elle atteint jusqu'à $t = 1800$ s lorsque $\alpha = 0,6$. D'après les courbes vertes, le dernier nœud opérationnel du réseau est mort à $t = 2450$ s lorsque $\alpha = 1,3$, tandis qu'il reste opérationnel jusqu'à $t = 2850$ s lorsque $\alpha = 0,6$. Sur la base de ces résultats, la valeur du paramètre α à une grande influence sur la fonctionnalité du réseau. Par conséquent, il est nécessaire de choisir sa valeur en fonction des besoins des consommateurs et des nécessités de l'application.

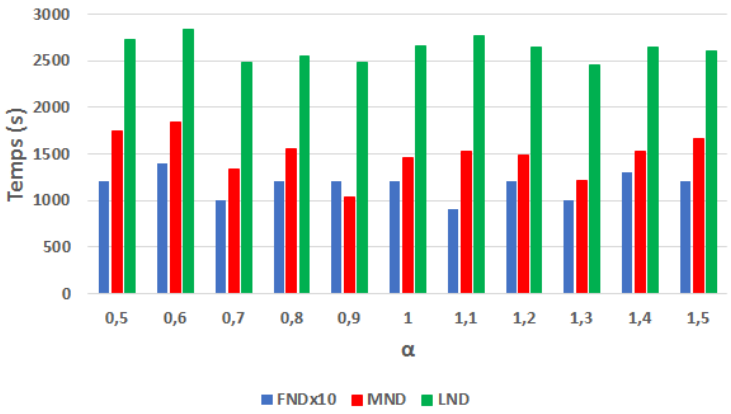


Figure 6.5 – FND, MND, et LND en variant le paramètre α

La figure 6.6 représente la durée de vie du réseau lors de l'utilisation de l'algorithme MSSA en variant le

paramètre α. La courbe montre que la meilleure durée de vie est obtenue lorsque la valeur de α est égale à $0,5$, $0,6$ ou $1,1$. Dans les environnements hostiles où le chargement ou le changement de batteries est presque impossible, le facteur énergétique est le plus pris en compte lors de la conception des protocoles réseau. Par conséquent, le paramètre α doit être défini sur $0,5$, $0,6$ ou $1,1$ pour un réseau opérant pour une longue période de temps.

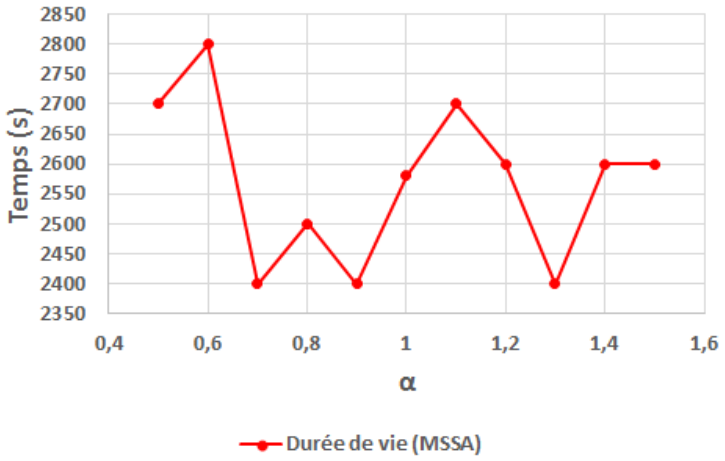


Figure 6.6 – Durée de vie en variant le paramètre α

La figure 6.7 représente le nombre de paquets de données délivrés au collecteur de données en variant le paramètre α dans l'algorithme MSSA. Nous notons que le meilleur taux de paquets délivrés est obtenu lorsque $\alpha = 1,1$. Dans les applications temps réel critiques où la contrainte de temps et l'urgence des données sont primordiales, il est préférable de définir la valeur de α égale à $1,1$.

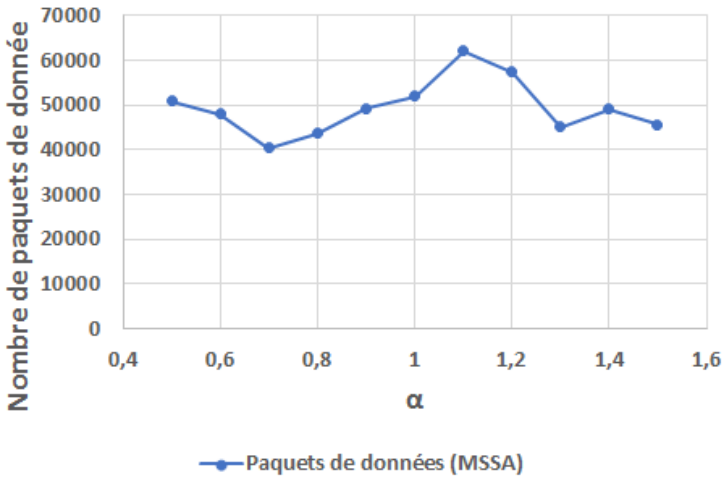


Figure 6.7 – Nombre de paquets de données délivrés en variant le paramètre α

6.2.6 MSSA vs SSA vs SMBFOA

La figure 6.8 représente le nombre de nœuds vivants en utilisant les trois algorithmes SMBFOA, SSA et MSSA avec $\alpha = 0,6$ et $\alpha = 1,1$. D'après les courbes, nous constatons que la durée de vie du réseau est légèrement augmentée dans le cas où l'algorithme MSSA est utilisé, une différence de 600 s est ajoutée. En outre, nous pouvons observer que lors de l'utilisation de la valeur 0.6 pour le paramètre α, la durée de vie du réseau est augmentée de 100 par rapport à un réseau qui utilisait la valeur de 1.1 pour le paramètre α. Cela peut s'expliquer par le fait que la complexité de l'algorithme MSSA est moindre que celle de l'algorithme SMBFOA. De plus, le paramètre α ajoute une certaine vitesse de convergence et d'efficacité pour trouver la position idéale pour le collecteur de données.

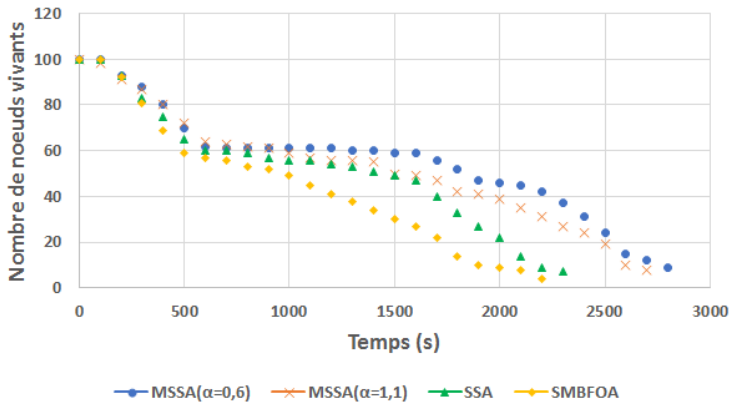


Figure 6.8 – Durée de vie de MSSA vs SSA vs SMBFOA

La figure 6.9 représente le nombre de paquets de données reçus par le collecteur de données en utilisant les trois algorithmes SMBFOA, SSA et MSSA avec $\alpha = 0,6$ et $\alpha = 1,1$. Notons que la quantité de données est meilleure, $\frac{1}{3}$ de paquets reçus en plus lors de l'utilisation de l'algorithme MSSA. Nous observons également que, lors de l'utilisation de $\alpha = 1,1$, la quantité de données reçues par le collecteur de données augmente de $\frac{1}{5}$ par rapport à celle observée lorsque $\alpha = 0,6$. Cela peut être justifié par le fait que cet algorithme utilise les paramètres c_1 qui équilibre entre l'exploration et l'exploitation et α qui accélère la vitesse de convergence pendant la recherche pour trouver la nouvelle position du collecteur de données.

La figure 6.10 représente l'énergie consommée par les nœuds en utilisant les trois algorithmes SMBFOA, SSA et MSSA avec $\alpha = 0.6$ et $\alpha = 1.1$. D'après les courbes, nous notons que à tout instant t, l'énergie consommée par le réseau utilisant l'algorithme MSSA est toujours inférieure à celle consommée par les deux autres algorithmes. MSSA trouve la meilleure solution en un minimum de temps. La position du collecteur de données étant optimisée, l'énergie utilisée par les points de rendez-vous (Chefs) lors de la transmission des données est réduite.

La figure 6.11 représente la distance séparant le collecteur de données des chefs obtenue en utilisant les trois algorithmes SMBFOA, SSA et MSSA avec $\alpha = 0,6$ et $\alpha = 1,1$. Nous notons que le collecteur de données est plus proche de $\frac{3}{5}$ des chefs utilisant MSSA lorsque $\alpha = 0,6$, où seuls $\frac{2}{5}$ des chefs sont proches

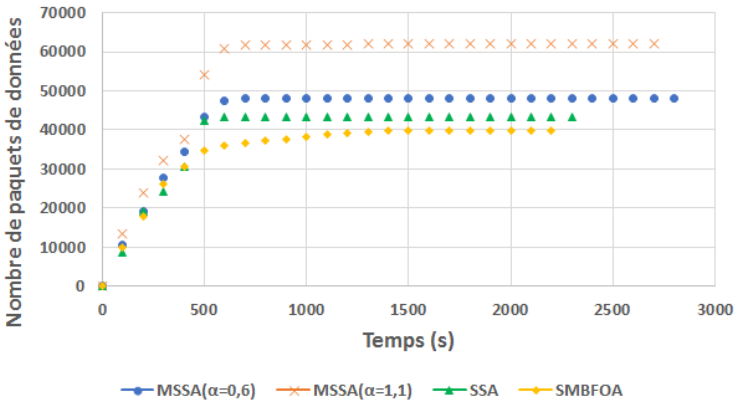


Figure 6.9 – Nombre de paquets de données reçus de MSSA vs SSA vs SMBFOA

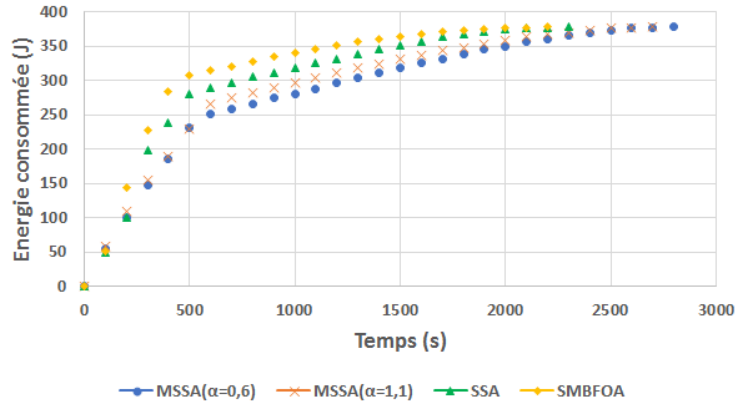


Figure 6.10 – Consommation énergétique de MSSA vs SSA vs SMBFOA

du collecteur de données lorsque $\alpha = 1,1$. Nous remarquons également que $\frac{3}{5}$ des chefs sont plus proches du collecteur de données en utilisant l'algorithme MSSA en comparant avec son homologue SMBFOA où seuls $\frac{2}{5}$ des chefs sont proches du collecteur de données. Un seul chef est éloigné du collecteur de données lors de l'utilisation de l'algorithme MSSA par rapport au SSA d'origine, où $\frac{4}{5}$ sont éloignés. L'algorithme MSSA trouve les meilleures coordonnées pour le collecteur de données par rapport aux algorithmes SSA et SMBFOA, ce qui améliore considérablement la latence de livraison des données et le rend idéal pour les applications sensibles au facteur temps.

6.3 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle approche de mobilité inspiré du mouvement des salpes nommé Modified salp swarm algorithm (MSSA), pour le choix du trajet du collecteur de données.

Cette nouvelle méthode est testée dans un réseau de topologie hétérogène dans lequel plusieurs itérations sont utilisées pour évaluer les performances de l'algorithme.

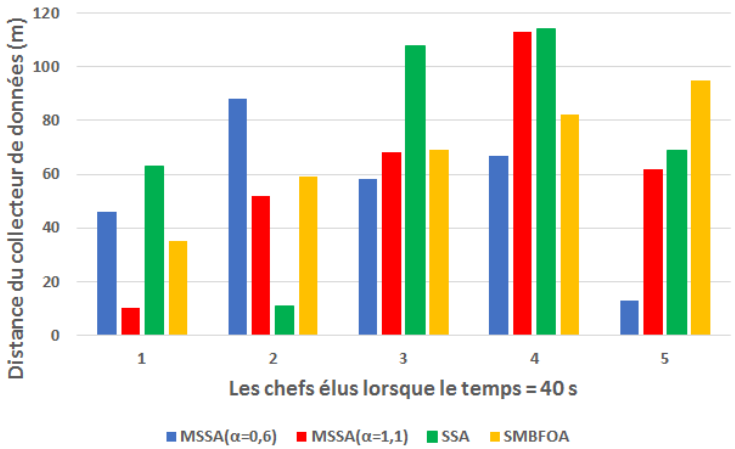


Figure 6.11 – Distance entre le collecteur de données et les chefs lorsque le $temps = 20$ s dans MSSA vs SSA vs SMBFOA

Les résultats de la simulation montrent que la collecte de données est meilleure lorsque l'approche proposée est adoptée avec succès. En outre, le collecteur de données effectue la tâche de routage de sorte que le chef préserve son énergie, ce qui se traduit par une amélioration de la durée de vie totale du réseau. De plus, la perte de données est minimisée et ainsi le problème de surcharge du tampon est également minimisé. L'intégration de ce routage intelligent améliore considérablement la collecte de données.

Le temps de collecte de données, le nombre d'itérations et la valeur du paramètre α jouent un rôle crucial dans l'amélioration des performances du réseau. Une période de collecte de données minimale permettra au collecteur de données de visiter rapidement tous les chefs des groupes. Un nombre d'itérations maximale augmentera la complexité de l'algorithme. La valeur du paramètre α dépendra du type d'application choisie et des caractéristiques à optimiser définies.

Par rapport à l'algorithme d'origine SSA et à l'algorithme SMBFOA inspiré des mouvements des bactéries, les résultats de simulation démontrent que la méthode proposée produit de meilleurs résultats par rapport aux autres algorithmes en termes de durée de vie et de débit.

Chapitre 7

Collecte de données urgentes dans un réseau hétérogène

Sommaire

7.1 Solution pour le débordement du tampon au niveau des chefs . . . 77
7.2 Évaluation des performances de CGMSSA 83
7.3 Conclusion 86

Dans ce chapitre, nous présentons l'algorithme Gravity center with modified salp swarm algorithm (CGMSSA) [88], un schéma de collecte de données urgentes intelligent qui prend en considération le dépassement du tampon au niveau des nœuds chefs dans un réseau de topologie hétérogène.

7.1 Solution pour le débordement du tampon au niveau des chefs

Dans cette section, une collecte de données optimisée évitant le débordement du tampon basée sur l'algorithme MSSA est présentée. Chaque groupe élit un chef en utilisant GWO [69]. Ensuite, nous planifions un chemin de déplacement optimal inspiré par le mouvement des salpes en fonction de l'heure d'arrivée du collecteur de données mobile et du temps de dépassement de tampon des points rendez-vous. Dans notre proposition, les nœuds de chaque groupe transfèrent les données à leur chef désigné, un collecteur de données mobile se positionne au centre de gravité du polygone formé par les chefs puis ajuste son emplacement pour la collecte de données en s'inspirant de l'essaim des salpes. L'organigramme de la solution proposée est illustré dans la figure 7.1.

7.1.1 Phase de regroupement

La phase de regroupement, élection des chefs et adhésion des membres, est décrite en détail dans le chapitre 5, page 40.

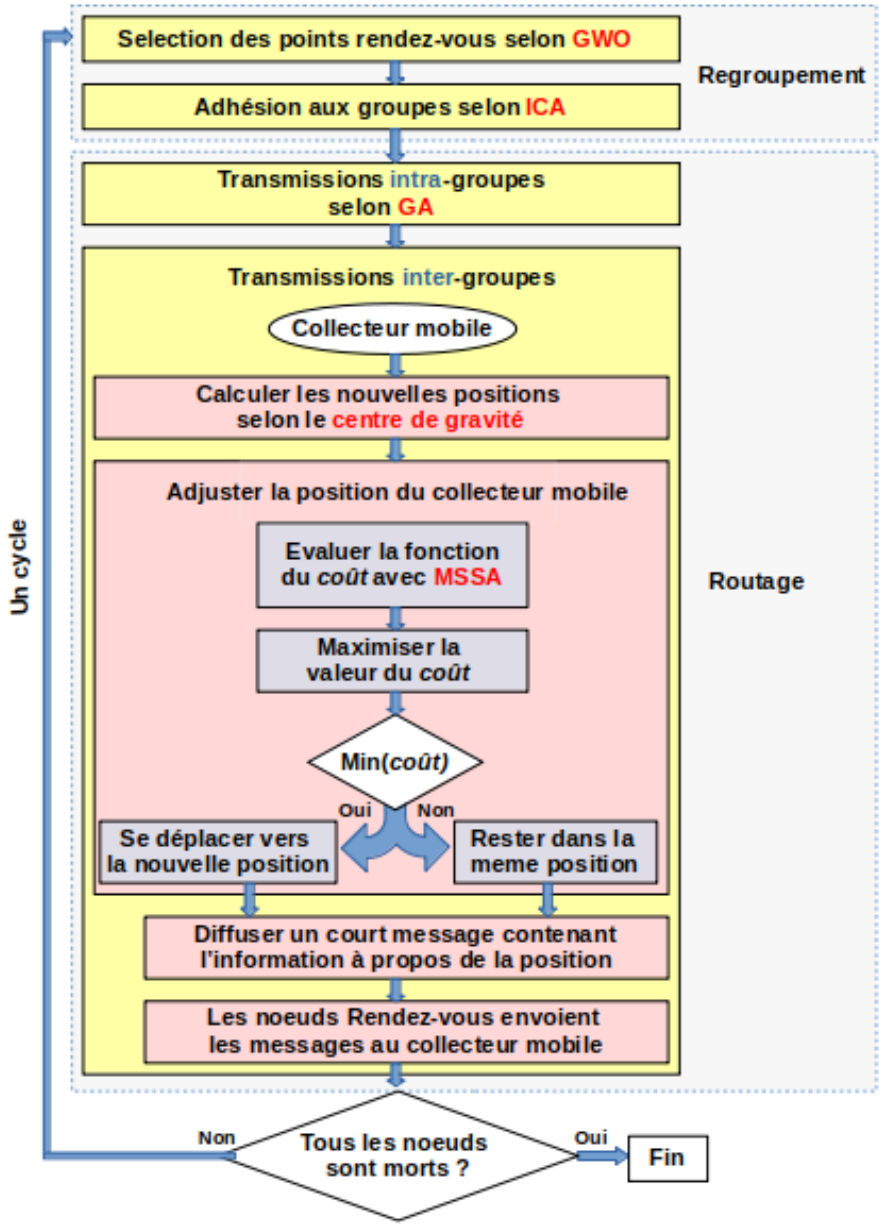


Figure 7.1 – Organigramme du protocole CGMSSA

7.1.1.1 Élection des chefs et adhésion aux groupes

La figure 7.2 expose de grands cercles colorés représentant les chefs. Les petits cercles expriment les nœuds non chef et les séparations en bleu symbolisent la formation des groupes. La voiture au centre de la zone de détection est le collecteur de données mobile. Le processus de sélection considère une nouvelle fonctionnalité : la taille du tampon du capteur. Les groupes se forment de manière autonome et intelligente en imitant la procédure hiérarchique des loups gris dans la vie réelle. Chaque nœud membre construit une route appropriée à l'aide de l'algorithme Genetic algorithm (GA), puis il envoie ces données à son chef correspondant. Après l'élection des chefs, les capteurs non-chef choisissent le meilleur groupe à rejoindre selon l'algorithme impérialiste ICA.

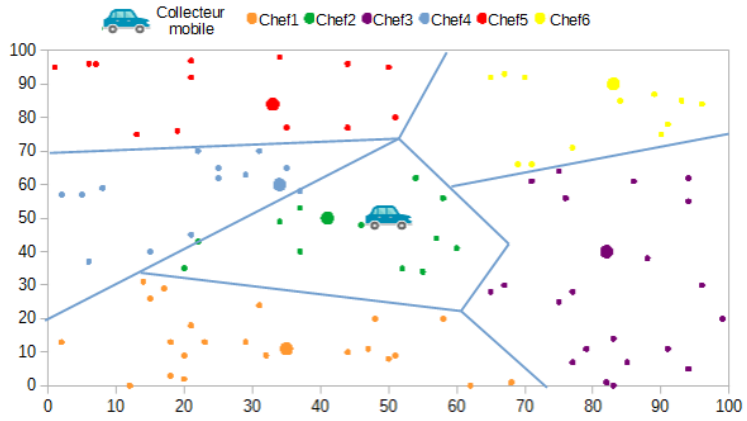


Figure 7.2 – Formation des groupes

7.1.2 Phase de routage

7.1.2.1 Transmissions intra-groupes

Après la phase de regroupement, le protocole LER-GA [80, 81] est utilisé pour transférer les données détectées du nœud à son chef associé. Le groupe est partitionné en zones et régions. Les nœuds du niveau inférieur envoient leurs données à leur successeur jusqu'à ce qu'ils arrivent à un niveau supérieur. Après réception de toutes les données des membres par le chef, celles-ci seront transmises à la SB via le collecteur de données.

7.1.2.2 Transmissions inter-groupes

1. **Déplacement du collecteur de données mobile vers le centre de gravité du polygone formé par les chefs :**
   Pour ce faire, nous calculons d'abord l'espace formé par les chefs comme dans Eq. 7.1.

$$S = \frac{1}{2} \times \sum_{i=0}^{n-1} (x_i \times y_{i+1} - x_{i+1} \times y_i) \tag{7.1}$$

Ensuite, en utilisant la formule donnée dans Eq. 7.2 et Eq. 7.3, nous calculons le centre de gravité d'un polygone irrégulier.

$$G_x = \frac{1}{6S} \sum_{i=0}^{n-1} (x_i + x_{i+1})(x_i \times y_{i+1} - x_{i+1} \times y_i) \tag{7.2}$$

$$G_y = \frac{1}{6S} \sum_{i=0}^{n-1} (y_i + y_{i+1})(x_i \times y_{i+1} - x_{i+1} \times y_i) \tag{7.3}$$

Prenons un exemple pour mieux comprendre le fonctionnement de l'algorithme. Supposons que nous avons 5 chefs (A, B, C, D, E) positionnés respectivement à (1,2) (7,5) (4,3) (6,1) (3,1). Le tableau 7.1 illustre comment la valeur du centre de gravité est calculée. Les nouvelles coordonnées du collecteur de données mobile sont représentées par le point G. La figure 7.3 montre le nouvel emplacement du collecteur de données mobile.

Chefs	$\alpha = (x_i \times y_{i+1} - x_{i+1} \times y_i)$	$\beta = \alpha \times (x_i + x_{i+1})$	$\delta = \alpha \times (y_i + y_{i+1})$
A	$(1 \times 5) - (7 \times 2) = -9$	$-9 \times (1 + 7) = -72$	$-9 \times (2 + 5) = -63$
B	$(7 \times 3) - (4 \times 5) = 1$	$1 \times (7 + 4) = 11$	$1 \times (5 + 3) = 8$
C	$(4 \times 1) - (6 \times 3) = -14$	$-14 \times (4 + 6) = -140$	$-14 \times (3 + 1) = -56$
D	$(6 \times 1) - (3 + 1) = 2$	$2 \times (6 + 3) = 18$	$2 \times (1 + 1) = 4$
E	$(3 \times 2) - (1 + 1) = 4$	$4 \times (3 + 1) = 16$	$4 \times (1 + 2) = 12$
Total	$\sum \alpha = -16$	$\sum \beta = -167$	$\sum \delta = -95$

Table 7.1 – Valeur du centre de gravité dans l'exemple

$$S = \left| \frac{1}{2} \times (-16) \right| = 8 \tag{7.4}$$

$$G_x = \frac{1}{6 \times (-8)} \times (-167) = 3,48 \tag{7.5}$$

$$G_y = \frac{1}{6 \times (-8)} \times (-95) = 1,98 \tag{7.6}$$

2. Ajustement de la position selon MSSA :
Après avoir positionné le collecteur de données au centre de gravité du polygone formé par les chefs, nous ajustons son emplacement selon l'algorithme MSSA (voir chapitre 6, page 67) pour collecter le maximum de données. Le collecteur de données tente d'approcher le chef ayant un plus grand nombre de membres.

Le tableau 7.2 illustre l'ajustement de la position du collecteur de données mobile en utilisant l'algorithme MSSA dans chaque itération pour notre exemple. c_1, c_2 et c_3 sont les paramètres de l'algorithme MSSA (voir section 6.1, page 67). À partir de la cinquième itération, nous avons remarqué qu'il n'y a aucune modification dans les valeurs x et y. La nouvelle position du collecteur mobile est alors $(3, 3)$. La figure 7.4 illustre la relocalisation du collecteur de données mobile.

La figure 7.5 représente la transmission de données du collecteur de données mobile vers la SB où le collecteur de données est positionné en A (le tour des chefs par le plus court chemin), G (centre de gravité des chefs)

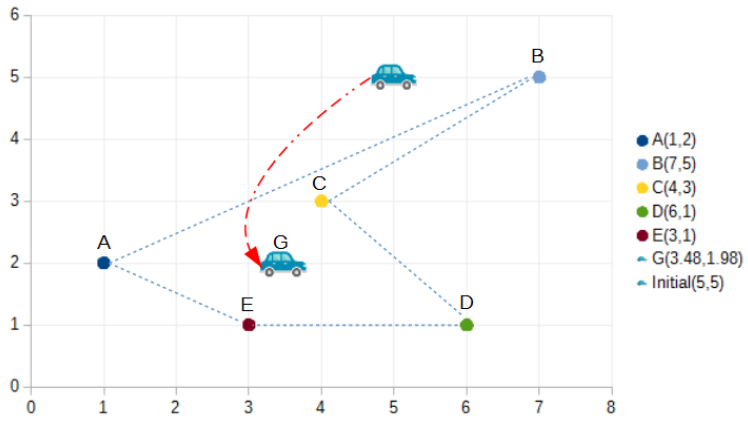


Figure 7.3 – Changement de la position du collecteur de données mobile

Itération	x	y	c_1	c_2	c_3
0	3,48	1,98	2	0,8973	0,6202
1	9	8	2	0,4625	0,5469
2	1	0	0.735759	0.2038	0.5916
3	4	4	0.036631	0.340500	0.340500
4	3	3	0.340500	0.340500	0.340500

Table 7.2 – Ajustement de la position du collecteur de données mobile dans l'exemple

et Salp (ajustement de la position par rapport au centre de gravité). "Distance du collecteur de données" correspond à la distance séparant le collecteur de données mobile et la SB. "Temps" est le temps passé à collecter les données de tous les chefs et "ET" est l'énergie dissipée pour faire cela. À partir des diagrammes, nous observons que la meilleure position pour le collecteur de données mobile, pour obtenir des résultats optimaux, est lorsqu'il est situé au centre de gravité des chefs. De meilleurs résultats sont obtenus lorsque cette position est ajustée par l'algorithme MSSA. Une différence remarquable se trouve lorsque le collecteur de données mobile est positionné dans le centre de gravité par rapport à celui lorsqu'il fait un tour d'un chef à un autre.
La figure 7.6 représente la transmission de données d'un collecteur de données mobile vers la SB positionné à l'intérieur et à l'extérieur du réseau. D'après les courbes, nous concluons que la position au centre de gravité donne toujours de meilleurs résultats même si la position de la SB est modifiée.

7.1.3 Traitement de l'urgence des données

Le collecteur de données mobile définit un identifiant pour chaque chef afin de classer l'urgence des données à collecter en tenant compte de la taille de son tampon libre. La valeur maximale définie correspond aux informations les plus urgentes avec l'espace libre minimal du tampon. L'urgence des données est classée en trois degrés : pas de tolérance au temps (valeur = 3), tolérance d'un certain temps (valeur = 2) et non important où la livraison des données peut attendre (valeur = 1). De même pour la taille de tampon libre, la valeur est classée en trois niveaux : la valeur 3 signifie qu'il n'y a pas de tampon libre, la valeur 2 signifie qu'il y a moins de 50% de tampon libre et la valeur 1 signifie que le chef en a plus de 50% du tampon libre. Ces deux valeurs sont additionnées pour déterminer l'identifiant du chef. Si plusieurs chefs ont la même valeur

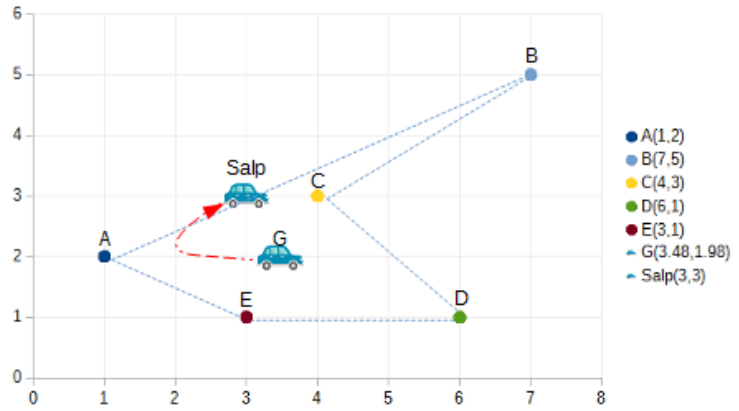


Figure 7.4 – Ajustement de la position du collecteur de données mobile

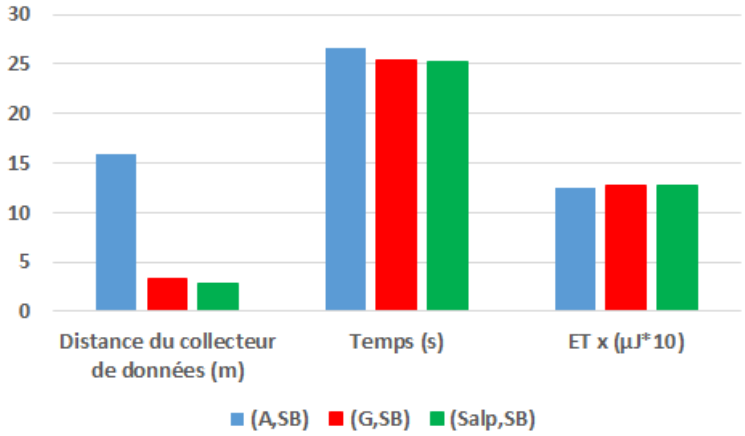


Figure 7.5 – Collecteur de données positionné à A, G et Salp

de leurs identifiants et la même valeur pour chacun des deux paramètres, urgence des données et la mémoire tampon libre, le chef le plus proche du collecteur de données mobile envoie ses données en premier. Si les chefs ont le même degré d'urgence, celui qui n'a pas d'espace envoie ses données en premier. Cependant, nous privilégions le chef intolérant au temps pour envoyer ses données en premier. La figure 7.7 montre le processus de classification des chefs par le collecteur de données mobile à l'aide de leurs identifiants. Les carrés bleus représentent le degré de tolérance de temps et les rouges représentent l'espace tampon libre. Ainsi, la classification du chef sera alors dans cet ordre : C, B, A, E et D. Lorsqu'un collecteur de données mobile termine une transmission de données pour un chef, il réinitialise sa valeur d'identification pour la prochaine session de collecte de données. Le meilleur classement du chef se fait ensuite en fonction de la tolérance de temps du chef et de son espace libre de tampon.

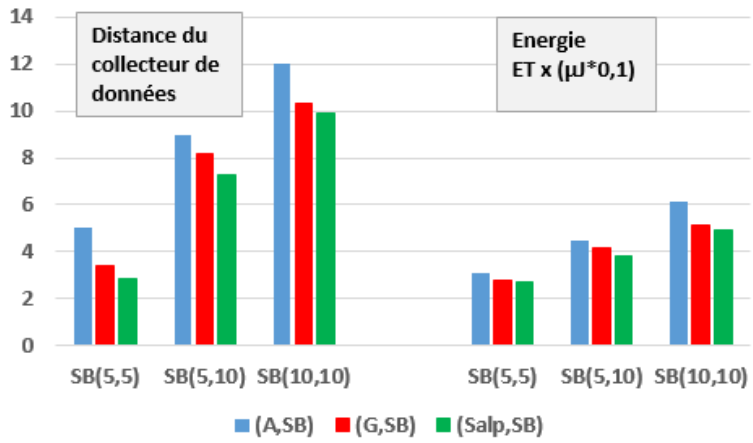


Figure 7.6 – SB positionnée à l'intérieur et à l'extérieur du réseau

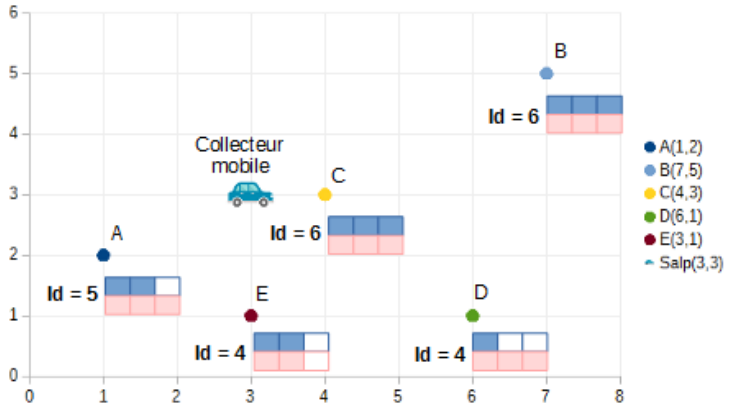


Figure 7.7 – Classification des chefs par le collecteur de données mobile

7.2 Évaluation des performances de CGMSSA

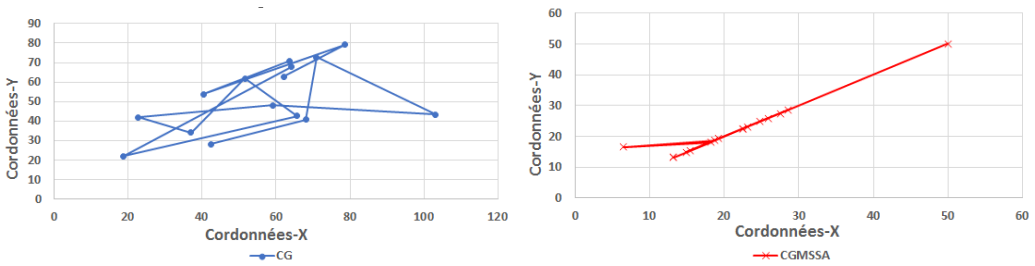
Pour évaluer les performances de l'algorithme proposé, une simulation a été réalisée. La zone de détection du réseau a été supposée être de 100 à 100 m^2 avec 100 nœuds hétérogènes déployé au hasard. Le collecteur de données est initialement positionné à (50 ; 50). Les nœuds envoient les données collectées à leurs chefs. Ensuite, ces chefs les redirigent vers le collecteur de données mobile. Les paramètres considérés pour les simulations sont donnés par le tableau 7.3.

7.2.1 Mouvement du collecteur de données

Les figures 7.8a et 7.8b représentent le mouvement du collecteur de données selon les algorithmes CG et CGMSSA. Nous remarquons d'après les parties a et b que le mouvement du collecteur de données est uniforme en utilisant CGMSSA contrairement au cas où l'algorithme CG est utilisé. Les positions du collecteur de données selon ce dernier sont dispersées.

Table 7.3 – Paramètres de simulation

Paramètres	Valeurs
Zone ciblée (m^2)	100 × 100
Position du collecteur de données	(50,50)
Nombre de nœuds	100
Énergie initiale du nœud (J)	2 ou 4
Transmetteur/Récepteur électroniques - E_{elec} (nJ/bit)	50
Amplificateur émetteur (espace libre)- ϵ_{fs} $(pj/bit/m^2)$	100
Amplificateur d'émetteur (multi-sauts) - ϵ_{mp} $(pJ/bit/m^4)$	0.013
Coût énergétique de l'agrégation des données - E_{DA} (nJ/bit)	50
Taille du paquet (bits)	4000
Nombre d'itérations	Variable
α	1.1
c_1,c_2	$\in$ [0 ; 1]



(a) Selon Gravity center (CG) (b) Selon CGMSSA

Figure 7.8 – Mouvement du collecteur de données

7.2.2 GWO vs CG vs CGMSSA

La figure 7.9 représente une comparaison en termes de durée de vie entre les algorithmes GWO [69], CG et CGMSSA. Nous remarquons que, le nombre de nœuds non fonctionnels s'élève au fur et à mesure que le temps s'accroit. Avant que temps = 500 s, cette augmentation est identique pour l'algorithme GWO et CG, par contre, en utilisant CGMSSA, 15 nœuds de plus restent fonctionnels. Lorsque le temps = 500 s, pratiquement la moitié des nœuds du réseau sont non fonctionnels. Entre 500 s et 2000 s, le nombre de nœuds non fonctionnels flambent lorsqu'il s'agit de l'algorithme GWO jusqu'à l'arrêt du réseau à l'instant 2000 s. Par contre, on observe une augmentation graduelle pour les deux autres algorithmes ou une variance de 13 nœuds, toujours opérateurs, est constaté lorsque CGMSSA est utilisé. Au-delà de 2000 s, la diminution des nœuds vivants est identique entre les deux algorithmes CG et CGMSSA. Donc, CG et CGMSSA augmente la durée de vie du réseau de $\frac{1}{3}$ comparée à GWO. Cette augmentation est due au fait que dans ces deux algorithmes, le collecteur de données est en mouvement intelligent. Il se positionne dans des endroits favorisant la conservation d'énergie et du coup prolongeant la durée de vie du réseau.

Les figures 7.10a et 7.10b représentent l'énergie consommée pour les algorithmes GWO [69], CG, et CGMSSA. De la partie a, nous remarquons qu'avant que le temps = 250 s, CGMSSA consomme plus d'énergie comparé à ces homologues GWO et CG. Passé cet instant, plus d'énergie a été consommée par GWO. De la partie b, CGMSSA a consommé 2 fois moins que GWO et la moitié de ce que CG a consommé dans une période de 500 s. Au-delà de 1500 s, les courbes CG et CGMSSA se chevauchent démontrant

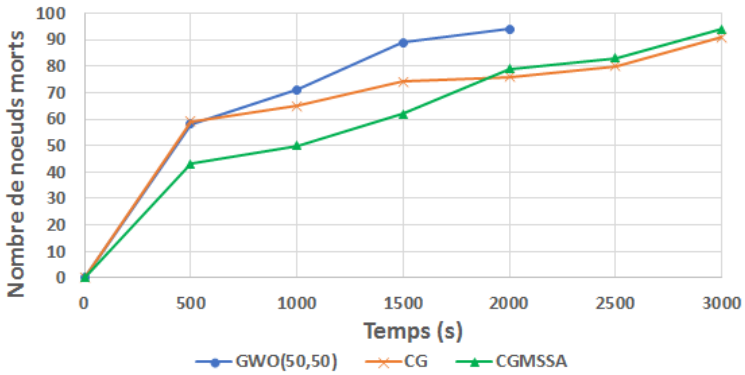
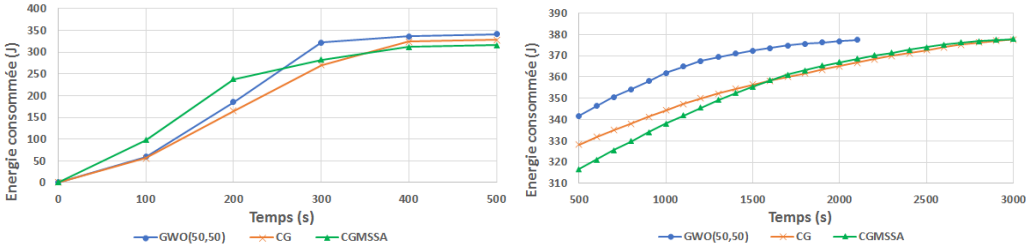


Figure 7.9 – Durée de vie de CG vs CGMSSA vs GWO

ainsi la même consommation d'énergie. Ces résultats peuvent être expliqués par le fait que l'incorporation du mouvement intelligent au collecteur de données qui se positionne stratégiquement entre les chefs leurs permettant de préserver leur énergie.



(a) Jusqu'à temps = 400 s (b) Jusqu'à temps = 3000 s

Figure 7.10 – Consommation énergétique de CG vs CGMSSA vs GWO

La figure 7.11 représente la quantité de données reçues par la SB en utilisant les algorithmes GWO [69], CG, et CGMSSA. Nous constatons que les courbes de GWO et CGMSSA se superposent pratiquement, sauf que l'émission de données pour GWO s'arrête à l'instant temps = 2000 s et celle de CGMSSA continue jusqu'à temps = 3000 s. L'algorithme CGMSSA produit la même quantité de données que GWO, qui équivaut à 25000 paquets, mais il a une durée de vie de 1000 s de plus par rapport à GWO. Pour ce qui concerne l'algorithme CG, avant temps = 1500 s, la réception de données au niveau de la SB est toujours moindre comparé aux autres algorithmes. Dépassé ce temps, la quantité de données reçues surpasse de 3350 paquets comparé à celle reçues en utilisant CGMSSA. Cette différence est due au fait que le temps que le collecteur de données ajuste sa position en utilisant MSSA, cette période est utilisée pour la réception de données pour l'algorithme CG. Puisque la position est amélioré avec la bio-inspiration, nous constatons une préservation d'énergie au niveau des chefs en utilisant CGMSSA.

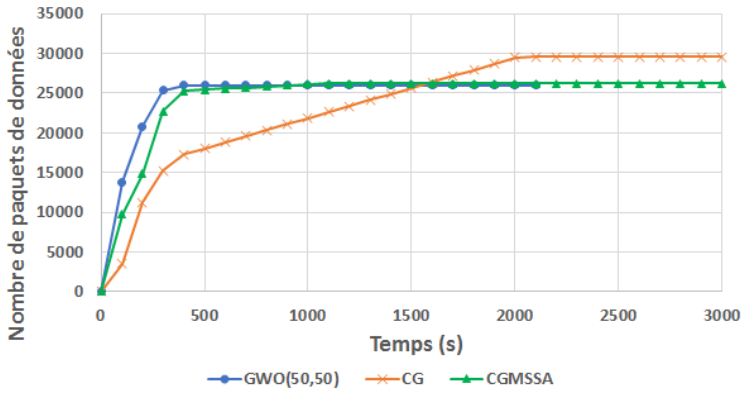


Figure 7.11 – Acheminement de données

7.3 Conclusion

Dans ce chapitre, nous avons étudié l'efficacité des protocoles hybrides bio-inspirés sur l'atténuation du débordement du tampon dans les réseaux temps-réels. Nous avons proposé une nouvelle relocalisation du collecteur de données en utilisant le centre de gravité (Gravity center). Les nouvelles coordonnées calculées par CG ont été ajustées par l'intelligence de l'essaim des salpes. En outre, l'urgence des données et la valeur de la mémoire tampon sont toutes les deux prises en compte pour concevoir le chemin du collecteur de données mobiles.

Les résultats de la simulation sont prometteurs. L'algorithme peut améliorer efficacement les performances globales du réseau, en terme d'énergie consommée et de quantité de données collectées.

Conclusion générale et perspectives

Conclusion générale

Dans cette thèse, nous avons traité de manière bio-inspirée le problème du débordement du tampon des capteurs, qui affecte considérablement les performances du réseau, lors de la collecte de données dans un RCSF hétérogène utilisant un collecteur de données mobile.

Dans un premier temps, nous avons présenté des généralités sur les RCSFs mobiles appliqués à l'IdO et les modèles heuristiques de mobilité assurant la collecte de données dans de tels réseaux. Puis, nous avons abordé la bio-inspiration appliquée aux choix des PRs et la mobilité du collecteur de données et nous avons présenté l'essentiel des travaux existants dans cet axe de recherche. Nous nous sommes concentrés sur le problème de la perte de données à cause du débordement du tampon des capteurs. A partir de ces deux techniques, nous avons proposé nos différentes contributions, résumées dans la figure 7.12.

Notre première contribution, SMBFOA, consiste en un nouveau modèle de mobilité inspiré par les bactéries afin de trouver le chemin optimal pour le collecteur de données tout en minimisant la perte de données dans un réseau homogène.

La deuxième contribution comporte deux approches de regroupements distribués s'inspirant de la communauté des loups gris et des baleines. Ils permettent de choisir idéalement les PRs visités par le collecteur de données mobile tout en minimisant la perte de données dans un réseau hétérogène.

La troisième contribution, MSSA, consiste en un nouveau modèle de mobilité inspiré par les salpes, réduisant la complexité de l'algorithme et applicable dans des réseaux de topologie hétérogène.

Dans la quatrième proposition, nous avons élaboré une solution qui peut être applicable dans une situation critique (données urgentes) en temps réel.

Les différentes contributions ont été évaluées en utilisant des simulations et ensuite elles ont été comparées avec des travaux similaires existants. Les résultats obtenus, résumés dans le tableau 7.4, sont très prometteurs en termes de collecte de données, consommation énergétique et complexité de calcul.

1. Prise en considération du dépassement de tampon

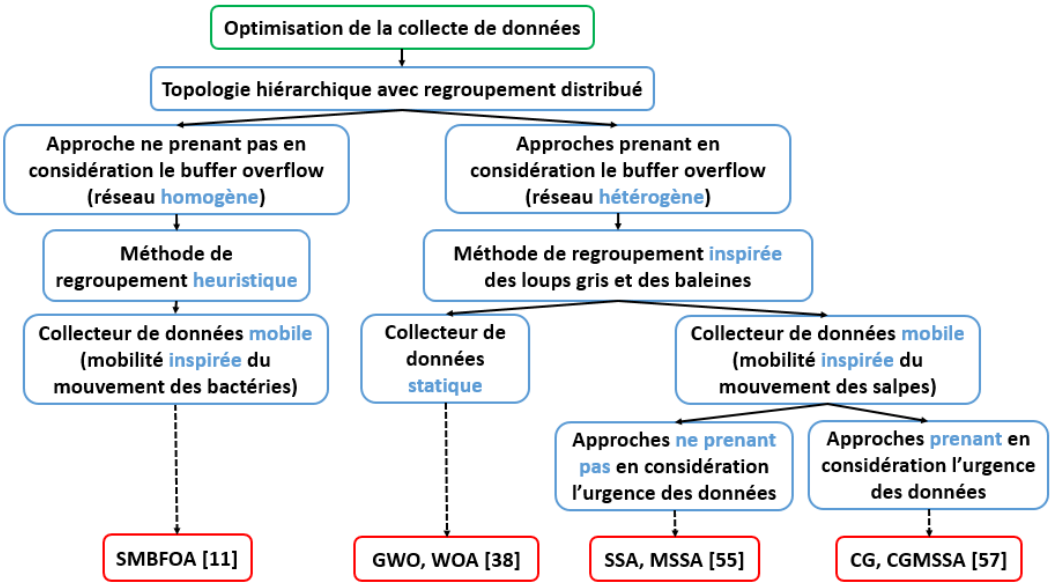


Figure 7.12 – Synthèses des contributions

Perspectives

Le but de la thèse était de proposer des solutions bio-inspirées pour la collecte de données dans les RCSFs mobiles appliqués dans un contexte IdO. Dans nos contributions, nous avons obtenu des résultats satisfaisants mais cela pourrait conduire à plusieurs améliorations dans différents niveaux. Nous énonçons quelques perspectives intéressantes qui pourraient améliorer ce travail.

Vitesse et durée de contact du collecteur de données : Le modèle de mobilité utilisé par le collecteur de données mobile affecte considérablement les performances de l'ensemble du réseau, non seulement le type de mobilité, mais également sa vitesse. Dans les cas où la vitesse du collecteur de données est élevée, il parcourt un espace considérable en très peu de temps, mais il aurait moins de temps pour collecter les paquets de données des capteurs dont il passe à proximité. Pour le cas de faibles vitesses, le contraire de cette situation est observé même s'il a suffisamment de temps pour recevoir tous les paquets de données des nœuds les plus proches. Les nœuds les plus éloignés doivent attendre davantage pour transmettre leurs données collectées lorsqu'il se rapproche.

Collecteurs de données multiples : La gestion des informations de localisation du collecteur de données augmente le nombre de paquets de contrôle. Lorsque l'emplacement change fréquemment, la surcharge du paquet de contrôle augmente, ce qui entraîne une consommation d'énergie plus élevée. Pour remédier aux problèmes susmentionnés, une solution viable consiste à augmenter le nombre de collecteurs de données et à partitionner la tournée totale de collecte en sous-chemins et affecter chaque collecteur à un sous-chemin [89]. La cœxistence de plusieurs collecteurs de données dans le réseau peut réduire les échecs d'interconnexion. Si un collecteur de données échoue, un autre peut jouer son rôle et transmet les données à sa place [45]. De plus, plusieurs collecteurs de données sont utilisés pour garantir la collecte périodique de données

Table 7.4 – Synthèse des résultats des contributions

Protocole	SMBFOA	GWO/WOA	MSSA	CG/CGMSSA
Regroupement	Heuristique	GWO/WOA	GWO	GWO
Collecteur de données	Mobile	Statique	Mobile	Mobile
Mobilité	BFOA	Non	MSSA	MSSA
Hétérogénéité	Non	Oui	Oui	Oui
Équilibrage de groupes	Non	Oui	Oui	Oui
Couverture	Non	Oui	Oui	Oui
Latence	Moyenne	Moyenne	Moyenne	Faible
Efficacité énergétique	Moyenne	Moyenne	Élevée	Élevée
Collecte de données	Faible	Moyenne	Élevée	Élevée
Complexité	Élevée	Moyenne	Faible	Faible
Dépassement de tampon [1]	Non	Oui	Oui	Oui
Urgence des données	Non	Non	Non	Oui
Comparaison	CDCMP	Fuzzy-GWO	SSA /SMBFOA	GWO

lorsqu'il s'agit de réseaux avec des objets mobiles [90]. À l'inverse, il sera très coûteux d'utiliser un nombre important de collecteurs de données, en particulier lorsqu'il s'agit de réseaux à grande échelle. Il semble très intéressant de trouver le nombre idéal des collecteurs de données à mettre dans un réseau.

Liste des publications personnelles

Revues internationales

- [11] R. Hamidouche, Z. Aliouat, A. M. Gueroui, A. A. A. Ari, L. Louail.
  Classical and bio-inspired mobility in sensor networks for iot applications. Journal of Network and Computer Applications 121 (2018) pages 70–88.
  Facteur d'impact 5.57
- [69] R. Hamidouche, Z. Aliouat, A. A. A. Ari, M. Gueroui.
  An efficient clustering strategy avoiding buffer overflowin iot sensors : A bio-inspired based approach. IEEE Access 7 (2019) pages 156733–156751.
  Facteur d'impact 4.098
- [86] R. Hamidouche, Z. Aliouat, A. A. A. Ari, M. Gueroui.
  Mobile sink path planning in heterogeneous IoT sensors : a salp swarm algorithm scheme. KSII Transactions on Internet & Information Systems, vol. 15, no. 6, pp. 2225-2239, 2021.
  Facteur d'impact 0.79
- [88] R. Hamidouche, Z. Aliouat, A. A. A. Ari, M. Gueroui.
  Optimized data gathering in a heterogeneous IoT network. International Journal of Communication Systems (2021).
  Facteur d'impact 1.319

Conférences internationales

- [7] R. Hamidouche, Z. Aliouat, M. Gueroui. Bio-inspired vs classical solutions to overcome the iot challenges, in : 2018 3^{rd} Cloudification of the Internet of Things (CIoT), IEEE, Paris, France, pp. 1–7.
- [12] R. Hamidouche, M. Khentout, Z. Aliouat, A. A. A. Ari, M. Gueroui. Sink Mobility based on Bacterial Foraging Optimization Algorithm, in : 6^{th} IFIP TC 5 International Conference, CIIA 2018, Oran, Algeria, May 8-10, 2018.
  Meilleur papier (2^{eme} place)

Autres publications

Revue internationale

- [80] R. Hamidouche, Z. Aliouat, M. Gueroui. Genetic algorithm for improving the lifetime and qos of wireless sensor networks, Wireless Personal Communications 101 (4) (2018) 2313–2348. **Facteur d'impact 1.2**

Conférence internationale

- [81] R. Hamidouche, Z. Aliouat, M. Gueroui. Low energy-efficient clustering and routing based on genetic algorithm in wsns, in : International Conference on Mobile, Secure, and Programmable Networking (MSPN), Vol. 11005, Springer, Paris, France, 2018, pp. 143–156.

Bibliographie

[1] K. Patel, C. Patel, S. S. Rizvi, and K. Elleithy, "An efficient approach to reduce the energy consumption in wireless sensor networks through active nodes optimization," in *Need For Engineering Education Conference*, A. S. F. E. Education, Ed. Honolulu, Hawaii : ASEE Annual Conference, April 2007, pp. 1–11.

[2] A. A. Taleb, T. Alhmiedat, O. A. Hassan, and N. M. Turab, "A survey of sink mobility models for wireless sensor networks," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 4, no. 9, pp. 679–687, September 2013.

[3] W. Liang, J. Luo, and X. Xu, "Prolonging network lifetime via a controlled mobile sink in wireless sensor networks," in *IEEE Global Telecommunications Conference (GLOBECOM'2010)*. Miami, FL, USA : IEEE, Decemeber 2010, pp. 1–6.

[4] Y. Derdour, B. Kechar, and F. Khelfi, "The impact of the mobile element on performance improvement in wireless sensor network," in *The 5^{th} International Conference on Ambient Systems, Networks and Technologies (ANT'2014)*, vol. 32. Hasselt, Belgium : Elsevier, June 2015, pp. 261–268.

[5] S. M. Hosseinirad, "Multi-layer clustering topology design in densely deployed wireless sensor network using evolutionary algorithms," *Journal of AI and Data Mining*, vol. 6, no. 2, pp. 297–311, 2018.

[6] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 38–43, 2017.

[7] R. Hamidouche, Z. Aliouat, and A. M. Gueroui, "Bio-inspired vs classical solutions to overcome the iot challenges," in *2018 3^{rd} Cloudification of the Internet of Things (CIoT)*. Paris, France : IEEE, 2018, pp. 1–7.

[8] N. N. Thilakarathne, M. K. Kagita, and W. M. Priyashan, "Green internet of things : The next generation energy efficient internet of things," in *Applied Information Processing Systems*. Springer, 2022, pp. 391–402.

[9] A. A. A. Ari, B. O. Yenke, N. Labraoui, I. Damakoa, and A. Gueroui, "A power efficient cluster-based routing algorithm for wireless sensor networks : Honeybees swarm intelligence based approach," *Journal of Network and Computer Applications*, vol. 69, pp. 77–97, 2016.

[10] A. Djedouboum, A. A. A. Ari, A. Gueroui, A. Mohamadou, and Z. Aliouat, "Big data collection in large-scale wireless sensor networks," *Sensors*, vol. 18, no. 12, pp. 44–74, 2018.

[11] R. Hamidouche, Z. Aliouat, A. M. Gueroui, A. A. A. Ari, and L. Louail, "Classical and bio-inspired mobility in sensor networks for iot applications," *Journal of Network and Computer Applications*, vol. 121, pp. 70–88, 2018.

[12] R. Hamidouche, M. Khentout, Z. Aliouat, A. M. Gueroui, and A. A. A. Ari, "Sink mobility based on bacterial foraging optimization algorithm," in *IFIP International Conference on Computational Intelligence and Its Applications*. Oran, Algeria : Springer, 2018, pp. 352–363.

[13] K. Georgiou, S. Xavier-de Souza, and K. Eder, "The iot energy challenge : A software perspective," *IEEE Embedded Systems Letters*, vol. 10, no. 3, pp. 53–56, 2017.

[14] S. Li, L. Da Xu, and S. Zhao, "The internet of things : a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[15] A. Ouaddah, H. Mousannif, A. Abou Elkalam, and A. A. Ouahman, "Access control in iot : Survey & state of the art," in *2016* 5^{th} *International Conference on Multimedia Computing and Systems (ICMCS)*. IEEE, 2016, pp. 272–277.

[16] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things : A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, June 2015.

[17] A. Whitmore, A. Agarwal, and L. Da Xu, "The internet of things—a survey of topics and trends," *Information systems frontiers*, vol. 17, no. 2, pp. 261–274, 2015.

[18] R. Van Kranenburg and A. Bassi, "Iot challenges," *Communications in Mobile Computing*, vol. 1, no. 1, pp. 1–5, 2012.

[19] M. C. Domingo, "An overview of the internet of things for people with disabilities," *journal of Network and Computer Applications*, vol. 35, no. 2, pp. 584–596, 2012.

[20] F. Al-Turjman and A. Radwan, "Data delivery in wireless multimedia sensor networks : Challenging and defying in the iot era," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 126–131, 2017.

[21] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, and P. Dutta, "The internet of things has a gateway problem," in *Proceedings of the 16th international workshop on mobile computing systems and applications*, 2015, pp. 27–32.

[22] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo, "A survey on facilities for experimental internet of things research," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 58–67, 2011.

[23] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[24] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks : a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[25] M. L. Kaddachi, A. Soudani, V. Lecuire, K. Torki, L. Makkaoui, and J.-M. Moureaux, "Low power hardware-based image compression solution for wireless camera sensor networks," *Computer Standards & Interfaces*, vol. 34, no. 1, pp. 14–23, 2012.

[26] H. Karl and A. Willig, *Protocols and architectures for wireless sensor networks*. John Wiley & Sons, 2007.

[27] A. Kinalis, S. Nikoletseas, D. Patroumpa, and J. Rolim, "Biased sink mobility with adaptive stop times for low latency data collection in sensor networks," *Information fusion*, vol. 15, pp. 56–63, 2014.

[28] P. Madhumathy and D. Sivakumar, "Enabling energy efficient sensory data collection using multiple mobile sink," *China Communications*, vol. 11, no. 10, pp. 29–37, 2014.

[29] E. B. Hamida, H. Ochiai, H. Esaki, P. Borgnat, P. Abry, and E. Fleury, "Measurement analysis of the live e ! sensor network : Spatial-temporal correlations and data aggregation," in *2009 Ninth Annual International Symposium on Applications and the Internet*. IEEE, 2009, pp. 263–266.

[30] B. Kechar and L. Sekhri, "Formal modelling and validation of a novel energy efficient cross-layer mac protocol in wireless multi hop sensor networks using time petri nets," in *2008 New Technologies, Mobility and Security*. IEEE, 2008, pp. 1–5.

[31] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of things architecture : Recent advances, taxonomy, requirements, and open challenges," *IEEE wireless communications*, vol. 24, no. 3, pp. 10–16, 2017.

[32] J. Luo and J.-P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3. IEEE, 2005, pp. 1735–1746.

[33] A. Bendjeddou, N. Ghoualmi-Zine, and C.-D. Pham, "Prolonging network lifetime and maintaining connectivity in mobile wireless sensor network," in *International Conference on Distributed System and Decision (ICDSD)*, vol. 2, Oran, Algeria, December 2014.

[34] R. Kacimi, R. Dhaou, and A.-L. Beylot, "Load balancing techniques for lifetime maximizing in wireless sensor networks," *Ad hoc networks*, vol. 11, no. 8, pp. 2172–2186, November 2013.

[35] S. Sharma, R. K. Bansal, and S. Bansal, "Issues and challenges in wireless sensor networks," in *2013 International Conference on Machine Intelligence and Research Advancement*. IEEE, 2013, pp. 58–62.

[36] T.-C. Hsu, Y.-W. P. Hong, and T.-Y. Wang, "Optimized random deployment of energy harvesting sensors for field reconstruction in analog and digital forwarding systems," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5194–5209, 2015.

[37] P. Balister and S. Kumar, "Random vs. deterministic deployment of sensors in the presence of failures and placement errors," in *IEEE INFOCOM 2009*. IEEE, 2009, pp. 2896–2900.

[38] A. Bendjeddou, N. Ghoualmi, and C. Pham, "A new method for prolonging network lifetime and maintaining the connectivity in wireless sensor network through controlling the transmission power," *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*, vol. 5, no. 1, pp. 1–14, 2014.

[39] A. Keerthika and V. B. Hency, "A survey of routing protocols of wireless sensor network with mobile sinks," *ARPN J. Eng. Appl. Sci.*, vol. 11, no. 11, p. 69516963, 2016.

[40] M. Maadi and M. Maadi, "Optimization of cluster heads selection by imperialist competitive algorithm in wireless sensor networks," *International Journal of Computer Applications*, vol. 89, no. 19, pp. 29–34, March 2014.

[41] S. Noritaka, M. Hiromi, M. Hiroki, and M. Michiharu, "Centralized and distributed clustering methods for energy efficient wireless sensor networks," *Lecture Notes in Engineering and Computer Science*, vol. 2174, pp. 978–988, March 2009.

[42] S. Kaur and R. N. Mir, "Clustering protocols in wireless sensor networks : A survey," *International Journal Computer Network and Information Security*, vol. 6, pp. 38–51, 2016.

[43] M. I. Khan, W. N. Gansterer, and G. Haring, "Static vs. mobile sink : The influence of basic parameters on energy efficiency in wireless sensor networks," *Computer communications*, vol. 36, no. 9, pp. 965–978, 2013.

[44] J. Wang, Y. Cao, B. Li, H.-j. Kim, and S. Lee, "Particle swarm optimization based clustering algorithm with mobile sink for wsns," *Future Generation Computer Systems*, vol. 76, pp. 452–457, 2017.

[45] A. Abuarqoub, M. Hammoudeh, B. Adebisi, S. Jabbar, A. Bounceur, and H. Al-Bashar, "Dynamic clustering and management of mobile wireless sensor networks," *Computer Networks*, vol. 117, pp. 62–75, 2017.

[46] A. A. Taleb, T. Alhmiedat, O. A.-H. Hassan, and N. M. Turab, "A survey of sink mobility models for wireless sensor networks," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 4, no. 9, pp. 679–687, 2013.

[47] L. Aliouat and Z. Aliouat, "Improved wsn life time duration through adaptive clustering, duty cycling and sink mobility," in *the* 8^{th} *International Conference on Information Management and Engineering (ICIME)*. Istanbul, Turkey : ACM New York, NY, USA, November 2016, pp. 36–40.

[48] Z. D. Shakir, K. Yoshigoe, and R. Lenin, "Adaptive buffering scheme to reduce packet loss on densely connected wsn with mobile sink," in *IEEE Consumer Communications and Networking Conference (CCNC)*. Las Vegas, NV, USA : IEEE, 2012, pp. 439–444.

[49] M. Faheem and V. C. Gungor, "Mqrp : Mobile sinks-based qos-aware data gathering protocol for wireless sensor networks-based smart grid applications in the context of industry 4.0-based on internet of things," *Future Generation Computer Systems*, vol. 82, pp. 358–374, 2018.

[50] A. Alsaafin, A. M. Khedr, and Z. Al Aghbari, "Distributed trajectory design for data gathering using mobile sink in wireless sensor networks," *AEU-International Journal of Electronics and Communications*, vol. 96, pp. 1–12, 2018.

[51] S. K. Mothku and R. R. Rout, "Adaptive buffering using markov decision process in tree-based wireless sensor and actor networks," *Computers & Electrical Engineering*, vol. 71, pp. 901–914, 2018.

[52] N. Sharma and V. Gupta, "Meta-heuristic based optimization of wsns energy and lifetime-a survey," in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2020, pp. 369–374.

[53] R. S. Priya and K. Arutchelvan, "Performance assessment of particle swarm optimization with grey wolf optimization algorithms for clustering in wireless sensor networks," in *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*. IEEE, 2020, pp. 505–509.

[54] B. M. Sahoo, H. M. Pandey, and T. Amgoth, "A whale optimization (woa) : Meta-heuristic based energy improvement clustering in wireless sensor networks," in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2021, pp. 649–654.

[55] M. Sharawi and E. Emary, "Impact of grey wolf optimization on wsn cluster formation and lifetime expansion," in 9^{th} *International Conference on Advanced Computational Intelligence (ICACI)*. Doha, Qatar : IEEE, 2017, pp. 157–162.

[56] P. Diwan and M. Khan, "Energy efficient communication for wsns using grey-wolf optimization algorithm," *International Journal Of Engineering And Computer Science*, vol. 5, no. 12, pp. 19 793–19 805, December 2016.

[57] X. Zhao, H. Zhu, S. Aleksic, and Q. Gao, "Energy-efficient routing protocol for wireless sensor networks based on improved grey wolf optimizer," *KSII Transactions on Internet & Information Systems*, vol. 12, no. 6, pp. 2644–2657, June 2018.

[58] A. R. Jadhav and T. Shankar, "Whale optimization based energy-efficient cluster head selection algorithm for wireless sensor networks," *arXiv preprint arXiv :1711.09389*, pp. 1–22, 2017. [Online]. Available : http://arxiv.org/abs/1711.09389

[59] M. Mehr, "Cluster head election using imperialist competitive algorithm (chei) for wireless sensor networks," *International Journal of Mobile Network Communications & Telematics*, vol. 4, no. 3, pp. 1–9, June 2014.

[60] Y. Jiang, L. Zhang, and J. Liu, "The path planning of mobile sink based on wolf pack algorithm," in *International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*. IEEE, 2019, pp. 147–150.

[61] D. Mechta and S. Harous, "Prolonging wsn lifetime using a new scheme for sink moving based on artificial fish swarm algorithm," in *Proceedings of the* 2^{nd} *International Conference on Advanced Wireless Information, Data, and Communication Technologies*. ACM, 2017, pp. 1–7.

[62] G. Yogarajan and T. Revathi, "Nature inspired discrete firefly algorithm for optimal mobile data gathering in wireless sensor networks," *Wireless Networks*, vol. 24, no. 8, pp. 2993–3007, April 2018.

[63] M. Krishnan, Y. M. Jung, and S. Yun, "An improved clustering with particle swarm optimization-based mobile sink for wireless sensor networks," in 2^{nd} *International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2018, pp. 1024–1028.

[64] A. A. A. Ari, I. Damakoa, A. Gueroui, C. Titouna, N. Labraoui, G. Kaladzavi, and B. O. Yenké, "Bacterial foraging optimization scheme for mobile sensing in wireless sensor networks," *International Journal of Wireless Information Networks*, vol. 24, no. 3, pp. 254–267, September 2017.

[65] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems*, vol. 22, no. 3, pp. 52–67, August 2002.

[66] H. Gu, A. Chen, X. Song, M. Brasch, J. Henderson, and D. Ren, "How escherichia coli lands and forms cell clusters on a surface : A new role of surface topography," *Scientific Reports*, vol. 6, p. 29516, 08 2016.

[67] R. V. Kulkarni, G. K. Venayagamoorthy, and M. X. Cheng, "Bio-inspired node localization in wireless sensor networks," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC2009)*. San Antonio, TX, USA : IEEE, October 2009, pp. 205–210.

[68] Y. Liu and K. Passino, "Biomimicry of social foraging bacteria for distributed optimization : models, principles and emergent behaviors," *Journal of Optimization Theory and Applications*, vol. 115, no. 3, pp. 603–628, December 2002.

[69] R. Hamidouche, Z. Aliouat, A. A. A. Ari, and M. Gueroui, "An efficient clustering strategy avoiding buffer overflow in iot sensors : A bio-inspired based approach," *IEEE Access*, vol. 7, pp. 156 733–156 751, 2019.

[70] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, March 2014.

[71] S. Zhang and Y. Zhou, "Grey wolf optimizer based on powell local optimization method for clustering analysis," *Discrete Dynamics in Nature and Society*, vol. 2015, pp. 1–17, October 2015.

[72] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer : a review of recent variants and applications," *Neural Computing and Applications*, vol. 30, no. 2, pp. 413–435, July 2018.

[73] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016.

[74] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm : An algorithm for optimization inspired by imperialistic competition," in *2007 IEEE congress on evolutionary computation*. Singapore : IEEE, 2007, pp. 4661–4667.

[75] N. Mittal, U. Singh, and B. S. Sohi, "Modified grey wolf optimizer for global engineering optimization," *Applied Computational Intelligence and Soft Computing*, vol. 2016, pp. 1–16, April 2016.

[76] K. Dudani and A. Chudasama, "Partial discharge detection in transformer using adaptive grey wolf optimizer based acoustic emission technique," *Cogent Engineering*, vol. 3, no. 1, pp. 1–15, November 2016.

[77] Z. Jabinian, V. Ayatollahitafti, and H. Safdarkhani, "Energy optimization in wireless sensor networks using grey wolf optimizer," *Journal of Soft Computing and Decision Support Systems*, vol. 5, no. 3, pp. 1–6, June 2018.

[78] S. Bandyopadhyay and E. J. Coyle, "Minimizing communication costs in hierarchically clustered networks of wireless sensors," *Computer Networks*, vol. 44, no. 1, pp. 1–16, 2004.

[79] N. Jain, P. Sinha, and S. Gupta, "Clustering protocols in wireless sensor networks : A survey," *International Journal of Applied Information Systems (IJAIS)*, vol. 5, no. 2, pp. 41–50, January 2013.

[80] R. Hamidouche, Z. Aliouat, and A. M. Gueroui, "Genetic algorithm for improving the lifetime and qos of wireless sensor networks," *Wireless Personal Communications*, vol. 101, no. 4, pp. 2313–2348, 2018.

[81] R. Hamidouche, Z. Aliouat, and A. Gueroui, "Low energy-efficient clustering and routing based on genetic algorithm in wsns," in *International Conference on Mobile, Secure, and Programmable Networking (MSPN)*, vol. 11005. Paris, France : Springer, 2018, pp. 143–156.

[82] A. Naeem, F. Azam, A. Amjad, and M. W. Anwar, "Comparison of model checking tools using timed automata - prism and uppaal," in *IEEE International Conference on Computer and Communication Engineering Technology (CCET)*. Beijing, China : IEEE, 2018, pp. 248–253.

[83] T. Fatima, K. Saghar, and A. Ihsan, "Evaluation of model checkers spin and uppaal for testing wireless sensor network routing protocols," in 12^{th} *International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. United States : IEEE, January 2015, pp. 263–267.

[84] L. Qing, Q. Zhu, and M. Wang, "Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks," *Computer Communications*, vol. 29, no. 12, pp. 2230–2237, 2006. [Online]. Available : http://www.sciencedirect.com/science/article/pii/S0140366406000727

[85] N. Al-Aboody and H. Al-Raweshidy, "Grey wolf optimization-based energy-efficient routing protocol for heterogeneous wireless sensor networks," in 4^{th} *International Symposium on Computational and Business Intelligence (ISCBI)*. Olten, Switzerland : IEEE, 2016, pp. 101–107.

[86] R. Hamidouche, Z. Aliouat, A. A. A. Ari, and A. Gueroui, "Mobile sink path planning in heterogeneous iot sensors : a salp swarm algorithm scheme," *KSII Transactions on Internet & Information Systems*, vol. 15, no. 6, pp. 2225–2239, 2021.

[87] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm : A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.

[88] R. Hamidouche, Z. Aliouat, A. M. Gueroui, and A. A. A. Ari, "Optimized data gathering in a heterogeneous iot network," *International Journal of Communication Systems*, vol. n/a, no. n/a, p. e4998, 2021. [Online]. Available : https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4998

[89] N. Kumar and D. Dash, "Mobile data sink-based time-constrained data collection from mobile sensors : a heuristic approach," *IET Wireless Sensor Systems*, vol. 8, no. 3, pp. 129–135, 2018.

[90] D. Dash, "Approximation algorithm for data gathering from mobile sensors," *Pervasive and Mobile Computing*, vol. 46, pp. 34–48, 2018.

Table des figures

1.1 Structure du marché mondial de l'IdO 7
1.2 Les types de collecteurs de données 10
1.3 Un graphe modélisant un réseau comportant 2 capteurs et un collecteur de données 11

2.1 Topologie des RCSFs mobiles 15
2.2 Classification des modèles heuristiques de mobilité 15
2.3 Le protocole DCMDC 16
2.4 Le protocole CDCMP 17
2.5 Le protocole FPM/SS 18
2.6 Le protocole MQRP 18
2.7 Les protocoles REP/RDP/DBP 19
2.8 Le protocole MDP 19

3.1 Classification des algorithmes bio-inspirés 23

4.1 Organigramme du protocole SMBFOA 33
4.2 Positions du collecteur de données avec la mobilité BFOA 35
4.3 Déploiement des nœuds 35
4.4 Nombre de paquets reçus par le collecteur de données et consommation énergétique à l'instant T = 460 s de SMBFOA vs CDCMP 36
4.5 Nombre de paquets de données reçus par le collecteur de données de SMBFOA vs CDCMP 37
4.6 Durée de vie de SMBFOA vs CDCMP 38
4.7 Consommation énergétique de SMBFOA vs CDCMP 38

5.1 Période d'exploration par rapport à la période d'exploitation en fonction de la valeur de A dans GWO 45
5.2 Adhésion aux groupes à l'aide d'ICA 52
5.3 Exemple illustratif 53
5.4 La transition d'état des protocoles GWO / WOA pour un cycle 54
5.5 Propriétés essentielles vérifiées sur UPPAAL 55
5.6 Diagramme de séquence de messages illustrant un seul cycle sur les protocoles GWO / WOA 56
5.7 Nombre d'itérations de GWO et WOA 59
5.8 Complexité temporelle de GWO et WOA 60

5.9 Différentes valeurs du paramètre a pour GWO et WOA . . . 61
5.10 Variation de la taille du réseau . . . 62
5.11 Les valeurs de la fonction de coût sur 20 exécutions indépendantes pour k, k + 5, k + 10 et k + 15 itérations . . . 62
5.12 GWO et WOA dans un réseau homogène et hétérogène . . . 63
5.13 Différents emplacements pour la SB pour GWO et WOA . . . 64
5.15 Différentes valeurs de $x(t+1)$ pour GWO et WOA . . . 65
5.16 Acheminement de données dans GWO vs WOA vs Fuzzy-GWO . . . 66

6.1 Notion du leader et du suiveur dans la chaîne des salpes . . . 68
6.2 Paquet diffusé par le chef après la fin de la phase de regroupement . . . 69
6.3 Variation de la durée de collecte de données . . . 71
6.4 Variation du nombre d'itérations . . . 72
6.5 FND, MND, et LND en variant le paramètre α . . . 72
6.6 Durée de vie en variant le paramètre α . . . 73
6.7 Nombre de paquets de données délivrés en variant le paramètre α . . . 73
6.8 Durée de vie de MSSA vs SSA vs SMBFOA . . . 74
6.9 Nombre de paquets de données reçus de MSSA vs SSA vs SMBFOA . . . 75
6.10 Consommation énergétique de MSSA vs SSA vs SMBFOA . . . 75
6.11 Distance entre le collecteur de données et les chefs lorsque le $temps = 20$ s dans MSSA vs SSA vs SMBFOA . . . 76

7.1 Organigramme du protocole CGMSSA . . . 78
7.2 Formation des groupes . . . 79
7.3 Changement de la position du collecteur de données mobile . . . 81
7.4 Ajustement de la position du collecteur de données mobile . . . 82
7.5 Collecteur de données positionné à A, G et Salp . . . 82
7.6 SB positionnée à l'intérieur et à l'extérieur du réseau . . . 83
7.7 Classification des chefs par le collecteur de données mobile . . . 83
7.8 Mouvement du collecteur de données . . . 84
7.9 Durée de vie de CG vs CGMSSA vs GWO . . . 85
7.10 Consommation énergétique de CG vs CGMSSA vs GWO . . . 85
7.11 Acheminement de données . . . 86
7.12 Synthèses des contributions . . . 88

A.1 Analyse statistique de projets de recherches heuristiques portant sur les principaux défis de l'IdO . . . 108
A.2 Analyse statistique de projets de recherches bio-inspirées portant sur les principaux défis de l'IdO . . . 108
A.3 Nombre de publications IEEE heuristiques et bio-inspirées sur l'IdO . . . 108

Liste des tableaux

2.1 Mobilité heuristique traitant le problème du dépassement de tampon 20
2.2 Bilan des approches de mobilité heuristiques 21

3.1 Tableau comparatif pour les approches bio-inspirées du regroupement des capteurs 26
3.2 Tableau comparatif pour les modèles de mobilité inspirés de l'intelligence en essaim . 27
3.3 Bilan des approches de mobilité bio-inspirées 27

4.1 Paramètres de simulation 36

5.1 Similitude entre GWO et le RCSF 42
5.2 Similitude entre WOA et le RCSF 48
5.3 Similitude entre ICA et le RCSF 50
5.4 Valeurs des différents paramètres dans l'exemple à t = 5ms 53
5.5 Paramètres de simulation 58

6.1 Signification des symboles 68
6.2 Paramètres de simulation 71

7.1 Valeur du centre de gravité dans l'exemple 80
7.2 Ajustement de la position du collecteur de données mobile dans l'exemple 81
7.3 Paramètres de simulation 84
7.4 Synthèse des résultats des contributions 89

Liste des algorithmes

Algorithme 5.1 Élection des chefs en utilisant GWO 43
Algorithme 5.2 Élection des chefs en utilisant WOA 49
Algorithme 5.3 Adhésion d'un membre à l'aide d'ICA 51
Algorithme 6.1 MSSA : Mouvement du collecteur de données 70

Liste des abréviations

IdO Internet des objets

WPA Path planning of mobile sink based on wolf pack algorithm

RCSF Réseau de capteurs sans fil

PDA Personal digital assistant

CCSF Capteur corporel sans fil

TIC Technologies de l'information et de la communication

RFID Radio frequency identification

SB Station de base

PSO Particle swarm optimization

GA Genetic algorithm

AIS Artificial immune system

BFOA Bacterial foraging optimization algorithm

QoS Quality of service

GPS Global position system

WATS Wide area tracking system

TDMA Time-division multiple access

SMBFOA Sink mobility based on bacterial foraging optimization algorithm

CDCMP Clustering duty cycle mobility aware protocol

TSP Traveling salesmen problem

NS Network simulator

ICA Imperialistic competition algorithm

GWO Grey wolf optimization

WOA Whale optimization algorithm

FIGWO Fitness value based improved grey wolf optimization

CHEI Cluster head election using imperialistic competition algorithm

CBS Current buffer size

IBS Initial buffer size

CBP Current battery power

IBP Initial battery power

LER-GA Low energy-efficient clustering and routing based on genetic algorithm

SSA Salp swarm algorithm

AFSA Artificial fish swarm algorithm

BI-DFAMDT Backbone integrated discrete firefly algorithm for mobile data transporter

MSSA Modified salp swarm algorithm

FND First node dead

MND Middle node dead

LND Last node dead

CG Gravity center

CGMSSA Gravity center with modified salp swarm algorithm

PR Point rendez-vous

DCMDC Dynamic clustering to efficiently maintain MDC relay networks

ZS Zone de service

REP Reduced energy path

RDP Reduced delay path

DBP Delay bound path

FPM Forwarding packet model with MS

SS Storing packet at source nodes

MDP Markov decision processus

SG Smart grid

MQRP Mobile sinks-based QoS-aware data gathering protocol

Troisième partie

Annexes

Annexe A

Étude statistique sur les approches heuristiques et bio-inspirées dans un contexte IdO

Nous présentons dans cette annexe une étude statistique sur le nombre d'articles publiés pour chaque défi IdO présenté dans le chapitre 1, page 6, de l'année 2010 à l'année 2018.

Pour les contributions heuristiques, nous nous sommes concentrés uniquement sur les articles publiés dans des revues. Les chercheurs s'intéressent particulièrement à la "gestion des données" (800 publications ces dernières années). Ensuite, ils étudient modérément la "conservation de l'énergie", la "mobilité", la "sécurité" et l'"hétérogénéité" (respectivement 490, 455, 426 et 277 articles) et n'accordent pas beaucoup d'importance aux autres défis (Figure A.1). Pour les contributions bio-inspirées, nous n'avons fait aucune exception, tous les types de papier sont comptés. Seuls les "problèmes de passerelle" et l'"interopérabilité" sont pris en compte par les chercheurs (respectivement 555 et 165 publications). La majorité des recherches bio-inspirées ne se concentrent pas sur la "mobilité des objets" ou la "sécurité des données", alors que ces défis sont très importants et les chercheurs doivent y accorder plus d'attention (Figure A.2).

La figure A.3a montre une quasi-absence de contributions bio-inspirées. La majorité des publications sont heuristiques. Cela peut s'expliquer par le fait que les chercheurs ne sont pas très alaises lorsqu'ils font des recherches en bio-inspiration. Ils sont obligés d'étudier la biologie, un domaine nouveau pour eux, pour établir une solution à leurs problèmes. Le nombre de publications continue d'augmenter au fil des années comme démontré dans la figure A.3b (12 publications en 2010 pour atteindre un nombre de 1597 publications en 2017).

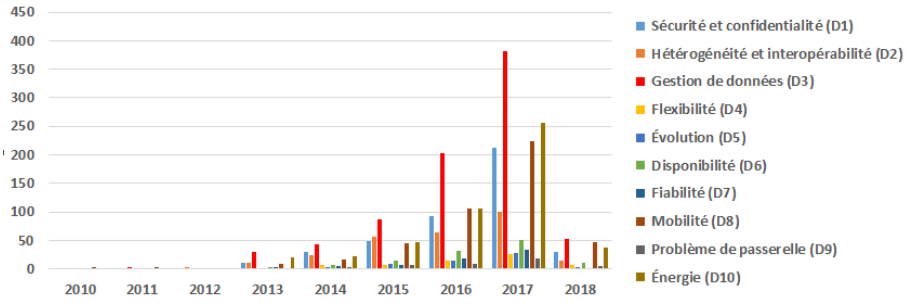


Figure A.1 – Analyse statistique de projets de recherches heuristiques portant sur les principaux défis de l'IdO

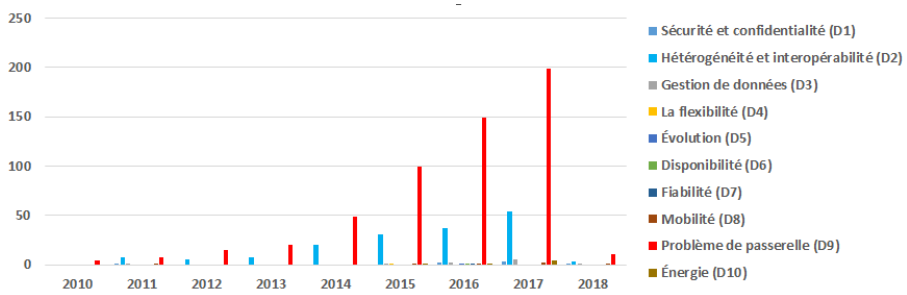
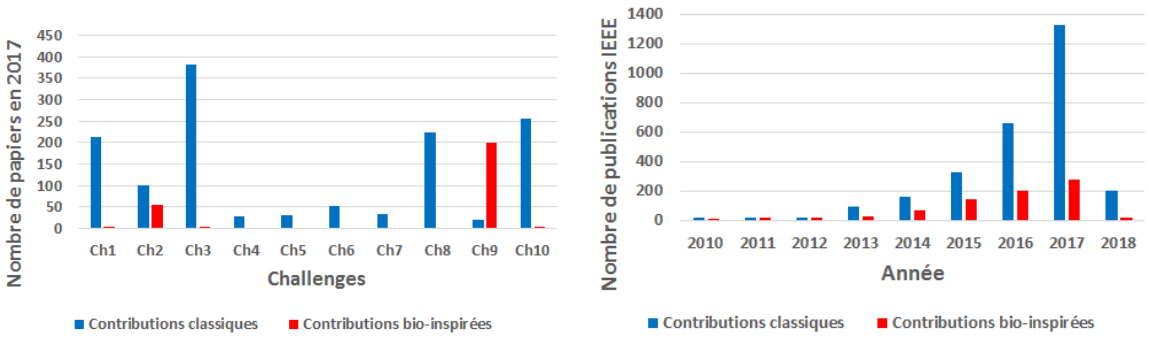


Figure A.2 – Analyse statistique de projets de recherches bio-inspirées portant sur les principaux défis de l'IdO



(a) En 2017

(b) Entre 2010 et 2018

Figure A.3 – Nombre de publications IEEE heuristiques et bio-inspirées sur l'IdO

Document réalisé avec LaTeXet :
le style LaTeXpour Thèse de Doctorat créé par F. Mazaleyrat – frederic.mazaleyrat@ens-paris-saclay.fr
la collection de paquets Modele_These_UParisSaclay_2021 peut être trouvée – ici