

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Faculté des Sciences Exactes
Département de Recherche Opérationnelle



Mémoire de fin de cycle

Pour l'Obtention du Diplôme de Master en Recherche Opérationnelle
Option : Modélisation mathématique et évaluation des performances des réseaux

Thème

OPTIMISATION DE LA CONSOMMATION D'ÉNERGIE DANS LES RÉSEAUX DE CAPTEURS SANS FIL AVEC UN PUIITS MOBILE

Présenté par

SADI KHAOULA
BOULDJENNET NESRINE

Soutenu le 08 juillet 2019 devant le jury composé de :

Melle. BOULFEKHAR Samra	MCA	Université de Béjaïa	Présidente
Dr. LARBI Ali	MCB	Université de Béjaïa	Examineur
Mme. BRAHMI Louiza	MAA	Université de Béjaïa	Examinatrice
Dr. BRAHMI Belkacem	MCA	Université de Béjaïa	Encadreur

Promotion 2018-2019

REMERCIEMENTS :

Tout d'abord, nous tenons à remercier Allah le tout puissant, à la sagesse et au savoir infinis, « Gloire à toi ! Nous n'avons de savoir que ce que Tu nous as appris. Certes c'est Toi l'Omniscient, le sage, le tout miséricordieux le très miséricordieux » (Sourate al-Baqarah, verset 32).

Nous tenons à remercier notre encadreur Mr BRAHMI pour le grand honneur qu'il nous a fait en nous proposant le sujet de ce mémoire de fin d'étude. Nous avons eu l'honneur et le privilège de travailler sous son assistance et de profiter de ses qualités humaine, professionnelle et de sa grande expérience, il nous a guidé tout au long de ce travail à l'élaboration de ce mémoire avec amabilité et dynamisme le caractérisant. Que ce modeste travail puisse satisfaire mes examinateurs, pour qu'ils en témoignent notre gratitude et reconnaissance pour l'aide et les conseils qu'il nous a prodigué, ainsi que pour le savoir qu'il nous a inculqué.

Nous remercions aussi tous nos enseignants de l'université Abd El Rahman Mira de Béjaia .

Nos remerciements vont également aux membres de jury d'avoir accepté de juger notre travail.

Nous remercions vivement nos familles et en particulier nos parents.

DÉDICACES :

Je dédie ce modeste travail à :

Mes très chers parents.

A mon mari Oussama.

A mes frères et ma soeur.

A ma meilleure amie Nouha et ma binôme Nesrine.

Mme khaoula,

Je dédie ce modeste travail à :

Mes très chers parents.

A mes frères et mes soeurs.

A toute la famille BOULDJENNET et BOUCEBCI sans exception.

A tout mes amies sans exception.

A tous les gens que j'aime.

Melle Nesrine,

TABLE DES MATIÈRES

Table des matières	3
Table des figures	6
Liste des tableaux	8
Liste des algorithmes	9
Liste des abreviations	10
Introduction Générale	12
1 Généralités sur les réseaux de capteurs sans fil	14
1.1 Introduction	14
1.2 Notions de base	14
1.2.1 Capteur	14
1.2.2 Réseau de capteurs sans fil	15
1.2.3 Station de base	16
1.3 Architecture d'un RCSF	16
1.4 Pile protocolaire	16
1.5 Classification des réseaux de capteurs sans fil	18
1.5.1 Type d'application	18
1.5.2 Mode de communication	20
1.5.3 Cluster	22

1.5.4	Mobilité des noeuds	23
1.5.5	Capacité des noeuds dans le réseau	24
1.6	Caractéristiques des réseaux de capteurs sans fil	24
1.7	Applications des RCSFs	26
1.8	Modèles de mobilité dans les RCSFs	27
1.8.1	La mobilité dans les RCSF	27
1.8.2	Architecture du RCSF mobile	27
1.8.3	Pourquoi la mobilité du puits ?	29
1.8.4	Les avantages de la mobilité du puits	29
1.8.5	Les enjeux liés à la mobilité	30
1.8.6	Les modèles de mobilité du puits dans les réseaux de capteurs	32
1.9	Position de problème	33
1.10	Conclusion	34
2	Problème de tournées	35
2.1	Introduction	35
2.2	Rappel sur l'optimisation combinatoire	35
2.2.1	Définition d'un problème d'optimisation combinatoire	35
2.2.2	Problèmes classiques d'optimisation combinatoire	36
2.2.3	Méthodes de résolution pour le PVC	38
2.3	Conclusion	50
3	Modélisation et résolution du problème	51
3.1	Introduction	51
3.2	Consommation d'énergie dans les RCSFs	52
3.2.1	Energie de capture	52
3.2.2	Energie de traitement	52
3.2.3	Energie de communication	52
3.3	Techniques de minimisation de la consommation d'énergie	53
3.4	Les principaux protocoles de routage dans les RCSF	54
3.4.1	Routage plat	54
3.4.2	Routage géographique	55
3.4.3	Routage hiérarchique	55
3.5	Méthodes du groupement	56
3.5.1	Méthode de K-moyennes	57

3.5.2	Le choix du cluster-head	57
3.5.3	Communication intra-cluster et inter-cluster	57
3.5.4	La maintenance des clusters	58
3.6	Protocole LEACH	58
3.7	Modèle de mobilité du puits proposé	61
3.7.1	Déscription du problème	61
3.8	Le modèle mathématique	61
3.9	Solution proposée	63
3.9.1	Hypothèses	63
3.9.2	Détail de la solution proposée	64
3.10	Conclusion	68
4	Expérimentation numérique	69
4.1	Introduction	69
4.2	Choix du simulateur	69
4.3	Métriques de simulation	70
4.4	Paramètres de simulation	71
4.5	Etapes de simulation	73
4.5.1	Initialisation des variables	74
4.5.2	Application de l'algorithme de clustering	74
4.5.3	Optimisation du chemin de routage	75
4.5.4	Affichage des résultats	76
4.6	Modèle de consommation d'énergie	76
4.7	Validation numérique	77
4.7.1	Les résultats obtenus	78
4.8	Evaluation de performances	78
4.8.1	Consommation d'énergie par tournée	79
4.8.2	Durée de vie du réseau	80
4.9	Conclusion	81
	Conclusion Générale	82
	Bibliographie	83

TABLE DES FIGURES

1.1	Capteur sans fil	15
1.2	Un réseau de capteurs sans fil	15
1.3	Architecture d'un RCSF.	16
1.4	Pile protocolaire d'un réseau de capteurs.	17
1.5	Type d'application orientée-temps	19
1.6	Type d'application orientée évènement	19
1.7	Type d'application orientée requête	20
1.8	Type de communication à un seul saut	21
1.9	Type de communication multi-sauts	21
1.10	Type de communication hiéararchisée	22
1.11	Type de communication multi-saut entre les clusters head	22
1.12	Exemple de structure de clusters	23
1.13	Applications des RCSF	27
1.14	Architecture du RCSFs mobile	28
2.1	Exemple de permutation 2-opt [42]	43
2.2	Minimums locaux et globaux d'une fonction	44
2.3	Illustration du codage des variables d'optimisation	47
2.4	Sélection par roulette	48
2.5	Le croisement en un point	49
2.6	Le croisement en deux points	49
2.7	Une mutation binaire	50

3.1	Les techniques de conservation d'énergie [30].	53
3.2	Topologie hiérarchique	56
3.3	Architecture du routage hiérarchique LEACH.	60
3.4	Schéma de la solution proposée.	65
4.1	Etapes de simulation.	73
4.2	Déploiement des noeuds.	74
4.3	Groupement des noeuds	75
4.4	La tournée de la station de base.	76
4.5	Modèle d'énergie utilisé [41].	77
4.6	Consommation d'énergie par tournée.	79
4.7	Durée de vie du réseaux en fonction de tournée.	80

LISTE DES TABLEAUX

4.1	Caractéristiques d'un noeud.	71
4.2	Paramètre de simulation.	72
4.3	Tableau comparatif entre LEACH et notre approche.	78

LIST OF ALGORITHMS

1	PPV.	40
2	K-means.	58
3	Phase d'installation.	65
4	Modèle de mobilité.	66
5	Phase de communication.	67

LISTE DES ABRIVIATIONS :

CH : Chef cluster.

LEACH : Low Energy Adaptive Clustering Hierarchical

LLC : Logical Link Control.

MAC : Médium Access Control.

MWSNs : mobile wireless sensor networks.

PPV : Plus proche voisin.

PVC : Problème du voyageur de commerce.

RCSFMs : Réseaux de capteurs sans fil mobiles.

RCSFs : Réseaux de capteurs sans fil.

SB : Station de base.

TSP : Traveling salesman problem.

WSNs : wireless sensor network.

INTRODUCTION GÉNÉRALE

Les progrès récents dans le domaine de la micro-fabrication et des technologies de communication sans fil, ont donné naissance à une nouvelle génération de réseaux, il s'agit des réseaux de capteurs sans fil (RCSFs).

Un réseau de capteurs sans fil est un type particulier des réseaux Ad-Hoc, il est constitué d'un grand nombre de dispositifs physiques appelés capteurs, pouvant relever des données de l'environnement surveillé (*température, vibration, mouvement des objets, etc*) et de les transmettre vers un centre de traitement. Les applications de ce type de réseaux sont très diverses, elles incluent le domaine militaire, le contrôle du trafic routier, les espaces sanitaires, le diagnostic industriel et autres.

Les capteurs sont dotés de batteries de taille très réduite, à faible autonomie et qui ne sont pas remplaçables. De ce fait, le réseau doit accomplir sa mission tout en conservant l'énergie des noeuds. Cette dernière est consommée dans trois fonctions : la capture, le traitement et la communication. Plusieurs facteurs interviennent dans ces fonctions, on cite : le type d'applications, le protocole de routage et la politique d'accès au canal de transmission, ...etc.

Le but d'un réseau de capteurs est défini par la capacité de faire router une grande quantité d'informations à une station de base, en assurant une longue période de vie du réseau. Le routage est l'une des problématiques majeures d'un réseau de capteurs sans fil, car la communication des données capturées est effectuée en sauts, jusqu'à ce que ces données arrivent à la destination.

L'objectif de notre travail est d'appliquer les techniques de la recherche opérationnelle dans le domaine des RCSFs avec une station de base mobile. On s'est intéressé particulièrement au problème de minimisation de la consommation d'énergie. Nous avons modélisé notre problème comme étant un problème du voyageur de commerce (PVC), où la station de base est considéré comme le voyageur, et les villes sont les chefs des clusters. La méthode utilisée pour résoudre ce PVC est l'heuristique du plus proches voisins (PPV).

Afin de valider notre proposition, nous avons implémenté notre solution sur MATLAB et nous l'avons comparé avec le protocole standard de LEACH [41] qui considère la station de base statique.

Ce mémoire est composé d'une introduction générale, quatre chapitres, d'une conclusion générale et d'une bibliographie :

- Le premier chapitre présente des généralités sur les réseaux de capteurs sans fil : leurs définitions, leurs architectures, leurs caractéristiques, et leurs domaines d'application. De plus, on décrit brièvement les différents modèles de mobilité proposés dans la littérature, et on termine par la positionnement du problème.
- Le deuxième chapitre est consacré aux rappels sur l'optimisation combinatoire et problème de voyageur de commerce. Nous avons aussi présenté les différentes approches de résolution du PVC, à savoir : la méthode de séparation et évaluation (Branch and Bound), l'heuristique PPV et la 2-optimalité, ainsi que les méta-heuristiques (recuit simulé et les algorithmes génétiques).
- Le troisième chapitre traite la modélisation du problème posé et propose un modèle de mobilité pour minimiser la consommation d'énergie dans les réseaux de capteurs sans fil tout en détaillant son principe et son fonctionnement.
- Le quatrième chapitre est consacré à simuler le protocole de routage LEACH et notre approche, sur MATLAB et faire une analyse des résultats obtenus.

Enfin, notre mémoire s'achève par une conclusion générale en résumant les grands points qui ont été abordés ainsi que des perspectives que nous souhaitons accomplir prochainement.

CHAPITRE 1

GÉNÉRALITÉS SUR LES RÉSEAUX DE CAPTEURS SANS FIL

1.1 Introduction

Un réseau de capteurs sans fil (RCSF) est un ensemble de noeuds déployés dans une zone de capture pour mesurer des grandeurs physiques, telles que : la température, la pression, la vibration,..., etc [21]. Les données collectées par ces capteurs sont envoyées à une station de base et acheminées par cette dernière vers l'utilisateur final via un autre réseau de communication. Ce type de réseau est utilisé par plusieurs applications militaires et civiles, telles que la santé, la surveillance, l'industrie, le transport,..., etc. Ces applications ont des exigences plus ou moins différentes, particulièrement, en termes de sécurité et conservation d'énergie.

Le but de ce chapitre est de présenter les RCSFs, décrire et introduire leurs architectures, leurs caractéristiques, leurs classifications, ainsi que leurs domaines d'applications. Ce chapitre s'achève par la position du problème et une conclusion.

1.2 Notions de base

1.2.1 Capteur

Un capteur sans fil est un petit dispositif électronique capable de mesurer une valeur physique environnementale (température, lumière, pression,..., etc.), et de la communiquer à

un centre de contrôle via une station de base [43].



FIGURE 1.1: Capteur sans fil

1.2.2 Réseau de capteurs sans fil

Un réseau de capteurs sans fil (RCSF) est un type particulier de réseau Ad-hoc [44], dans lesquels les noeuds sont des "capteurs". Ils se composent généralement d'un grand nombre de capteurs communicants entre eux via des liens radio pour le partage d'informations et le traitement coopératif des données. Dans ce type de réseau, les capteurs échangent des informations par exemple sur l'environnement pour construire une vue globale de la région contrôlée, qui est rendue accessible à l'utilisateur externe par un ou plusieurs noeuds. Les données collectées par ces capteurs sont acheminées directement ou via les autres capteurs de proche en proche à un "point de collecte", appelé station de base. Cette dernière peut être connectée à une machine puissante via Internet ou par satellite.

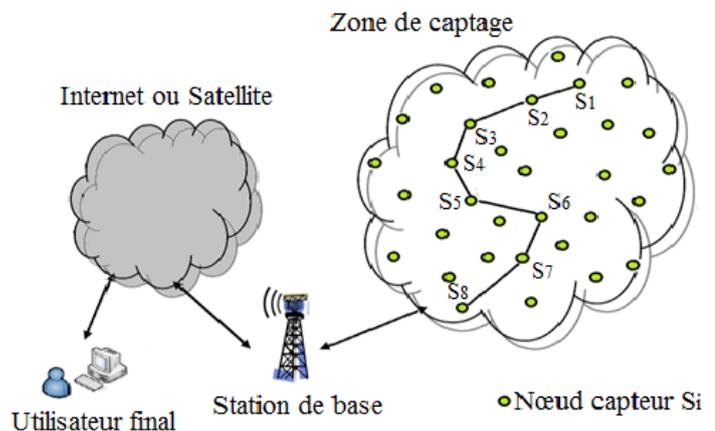


FIGURE 1.2: Un réseau de capteurs sans fil

1.2.3 Station de base

C'est un noeud particulier du réseau, qui est appelé aussi puits (sink). Il est chargé de la collecte des données issues des différents noeuds du réseau. Il doit être toujours actif, puisque l'arrivée des informations est aléatoire. Et c'est pourquoi son énergie doit être illimitée. Dans un RCSF plus ou moins large et à charge peu élevée, nous pouvons trouver deux puits ou plus pour alléger la charge [16].

1.3 Architecture d'un RCSF

Un RCSF est composé d'un ensemble de noeuds capteurs, organisés dans un champs de capture (sensor fields), comme le montre la figure 1.4. Chacun de ces noeuds a la capacité de collecter des données et de les transférer aux noeuds puits par l'intermédiaire d'une architecture multi-sauts. Le puits transmet ensuite ces données par Internet ou par satellite à l'ordinateur central (Gestionnaire de tâches) pour analyser ces données et prendre des décisions [44].

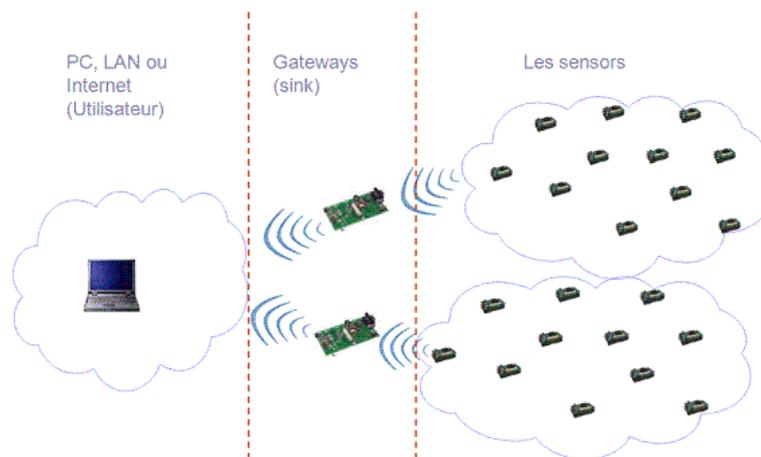


FIGURE 1.3: Architecture d'un RCSF.

1.4 Pile protocolaire

La pile protocolaire [43] utilisée par la station de base ainsi que tous les autres capteurs du réseau est illustrée par la figure 1.5. Elle comprend cinq couches, à savoir : la couche application, transport, réseau, liaison de données, physique, et trois plans de gestion : D'énergie, de mobilité et de tâches.

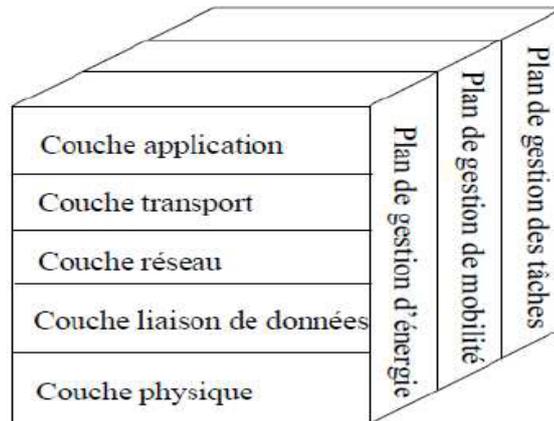


FIGURE 1.4: Pile protocolaire d'un réseau de capteurs.

Couche physique : assure la transmission et la réception des données au niveau bit.

Couche liaison de données : cette couche garantir une faible consommation d'énergie en minimisant le taux de collision entre les données diffusées par les noeuds voisins. Elle est composée d'une couche de contrôle des liens logiques (*LLC pour Logical Link Control*) et une couche de contrôle d'accès au médium (*MAC pour Médium Access Control*).

Couche réseau : cette couche assure l'opération de routage qui consiste à découvrir les chemins des données.

Couche transport : La couche transport aide à gérer le flux de données si le réseau de capteurs l'exige. Elle permet de diviser les données issues de la couche application en segments pour les délivrer, ainsi elle réordonne et rassemble les segments venus de la couche réseau avant de les envoyer à la couche application.

Couche application : cette couche permet à l'utilisateur d'interagir avec le réseau de capteurs à travers des interfaces. Il s'agit donc du niveau le plus proche des utilisateurs, géré directement par logiciels.

Plan de gestion d'énergie : Les fonctions intégrées à ce niveau consistent à gérer l'énergie consommée par les capteurs. Dès lors, un capteur peut par exemple éteindre son interface de réception dès qu'il reçoit un message d'un noeud voisin afin d'éviter la réception

des messages dupliqués. De plus, quand un noeud possède un niveau d'énergie faible, il peut diffuser un message aux autres capteurs pour ne pas participer aux tâches de routage, et conserver l'énergie restante aux fonctionnalités de capture [43].

Plan de gestion de mobilité : Ce niveau détecte et enregistre tous les mouvements des noeuds capteurs, d'une manière à leur permettre de garder continuellement une route vers l'utilisateur final, et le noeud peut garder trace de l'emplacement de ses voisins. En connaissant l'emplacement des voisins, les noeuds peuvent équilibrer l'usage de leur énergie [43].

Plan de gestion de tâche : Le niveau de gestion des tâches assure l'équilibrage et la distribution des tâches sur les différents noeuds du réseau, afin d'assurer un travail coopératif et efficace en matière de consommation d'énergie, et par conséquent, prolonger la durée de vie du réseau [43].

1.5 Classification des réseaux de capteurs sans fil

Les réseaux de capteurs sans fil sont des réseaux orientés application. De ce fait, leurs caractéristiques se diffèrent d'un domaine d'application à un autre. Ils se distinguent par le type d'application pour laquelle ils sont conçus, le mode de communication [45], la mobilité, et selon la capacité des noeuds dans le réseau [45].

1.5.1 Type d'application

Le mode de collecte et de livraison des données dans les réseaux de capteurs dépendent étroitement du type d'application. Ainsi, les applications des réseaux de capteurs sans fil peuvent être classés en quatre types d'applications : orientées temps (time driven), orientées événements (event driven), orientées requêtes (query driven) et hybrides [29].

– Applications orientées-temps

Dans ce type d'application, les capteurs font leur échantillonnage d'une manière périodique dans des intervalles de temps réguliers. Ensuite, ils envoient ces données captées à la station de base périodiquement, comme le montre la figure 1.6. Un exemple d'utilisation de ce type de réseau est la collecte des données environnementales (agriculture,

étude de phénomène naturel...).

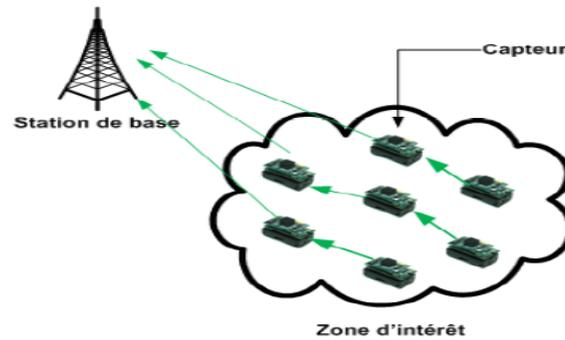


FIGURE 1.5: Type d'application orientée-temps

– Applications orientées évènements

Dans ce type d'applications, les capteurs envoient les données à la station de base seulement si un évènement spécial se produit. A titre d'exemple, l'évènement peut être la détection de la fumée, voir la figure 1.7. Ce type de réseau peut être appliqué dans différents domaines tels que la surveillance médicale (surveillance de taux de glycémie dans le sang), le contrôle d'édifice (les barrages, les voies des chemins de fer...) et la surveillance militaire,..., etc.

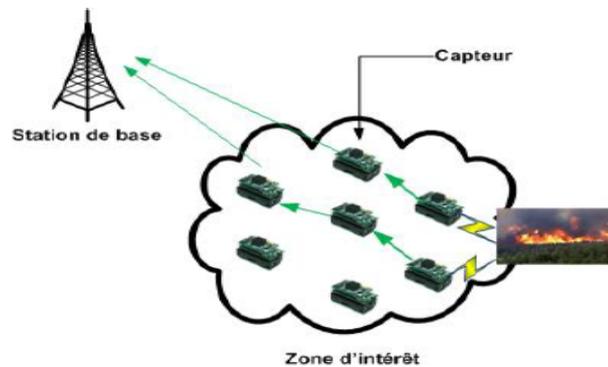


FIGURE 1.6: Type d'application orientée évènement

– Applications orientées requêtes

Dans les applications orientées requêtes, les capteurs font un échantillonnage à la demande de la station de base. Lorsqu'un capteur reçoit une requête de la part de la station de base, il déclenche la collecte de données. Après, il envoie ces données collectées à la station de base, figure 1.8. Cette catégorie de réseau est destinée aux applications adaptées à l'utilisateur (demande des informations sur une région bien précise). Dans ce type de réseau, la topologie et la position des noeuds doivent être connues.

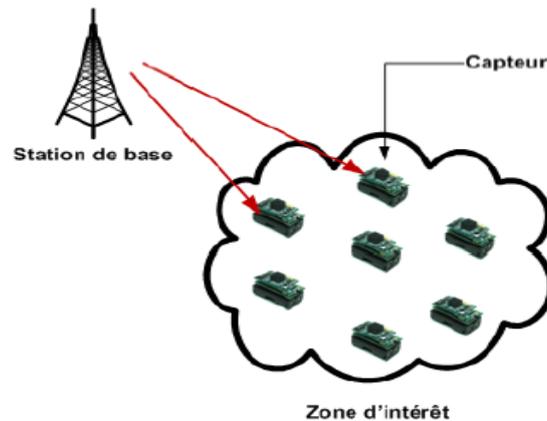


FIGURE 1.7: Type d'application orientée requête

– Application hybride

Le type d'application hybride est une combinaison entre les trois types d'applications précédemment décrites. Par exemple, dans un réseau conçu pour le suivi d'objets, le réseau peut combiner entre un réseau de surveillance (time-driven) et un réseau de collecte de données par événements (event-driven). Par exemple, pendant les longues périodes d'inactivité des capteurs et lorsqu'aucun objet n'est présent, le réseau peut assurer une fonction de surveillance.

1.5.2 Mode de communication

Le mode de communication utilisé dans le réseau de capteurs dépend du type d'application et des techniques utilisées pour faire acheminer l'information des capteurs à la station de base. On distingue dans cette classification trois types de communications :

– Communication à un seul-saut

Dans les réseaux de capteurs où la communication est à un seul-saut, les noeuds capteurs envoient les données collectées directement à la station de base sans passer par des noeuds intermédiaires. Dans ce type de réseau, les noeuds envoient leurs données en utilisant une forte puissance comme illustré dans la figure 1.9.

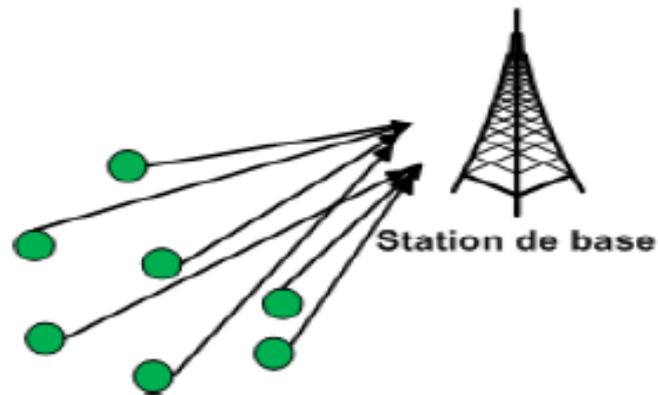


FIGURE 1.8: Type de communication à un seul saut

– **Communication multi-sauts**

Dans les réseaux de capteurs où la communication est multi-sauts, un noeud capteur envoie ses données à la station de base par l'intermédiaire de ses noeuds voisins en utilisant une petite puissance de transmission. La figure 1.10 illustre un exemple sur ce type de communication. Ce type de réseau est appliqué dans plusieurs domaines d'application.[\[20\]](#).

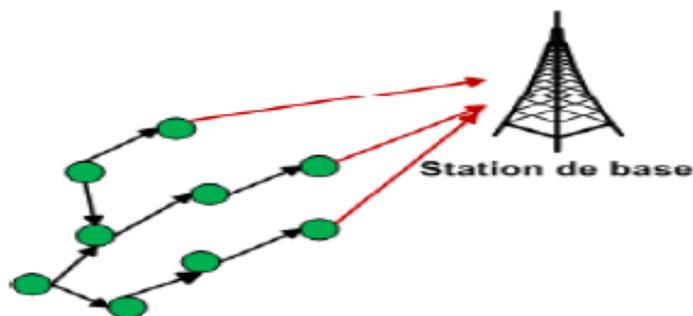


FIGURE 1.9: Type de communication multi-sauts

– Les réseaux de capteurs hiérarchisés

Dans les réseaux de capteurs hiérarchisés, la zone d'observation est divisée en clusters. Un cluster head est élu pour chaque cluster. Ce dernier s'occupe de récupérer les informations auprès des capteurs dans son cluster et de les transmettre directement à la station de base voir la figure 1.11 ou via un mode multi-saut entre les clusters head voir la figure 1.12.

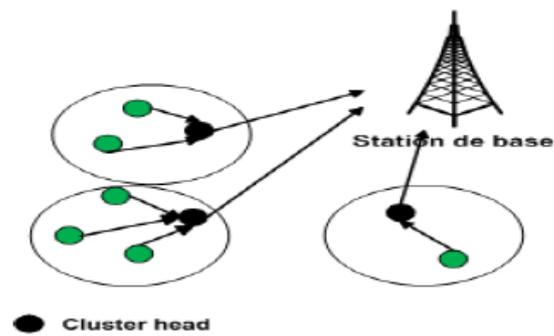


FIGURE 1.10: Type de communication hiérarchisée

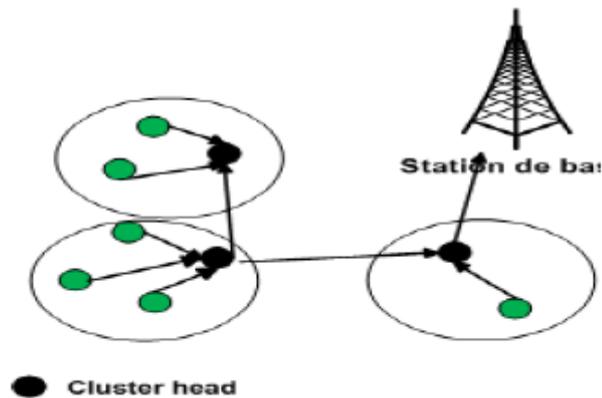


FIGURE 1.11: Type de communication multi-saut entre les clusters head

1.5.3 Cluster

Un cluster est un sous-ensemble de noeuds connexe du réseau, et la structuration où le clustering est le processus de regroupement des noeuds en clusters disjoints [44]. Comme le

montre la figure 1.3, généralement les clusters comportent trois types de noeuds :

- *Chef de cluster* : est un noeud particulier, appelé Cluster Head (CH). Ce dernier, permet de coordonner les membres de son cluster, d'agréger et /ou de traiter les données collectées et de les transmettre à la station de base. Le chef de cluster est choisi pour jouer ce rôle soit d'une manière déterministe (chef de cluster prédéfini) ou d'une manière aléatoire (chef de cluster élu parmi les noeuds du réseau selon une métrique bien particulière ou une combinaison de métriques).
- *Noeud passerelle (gateway)* : possède des liens inter-clusters et peut donc accéder à des clusters voisins et acheminer les données entre eux.
- *Noeud ordinaire* : il ne possède pas de liens avec les autres clusters et quand il s'attache à un chef de cluster en devenant membre du cluster.

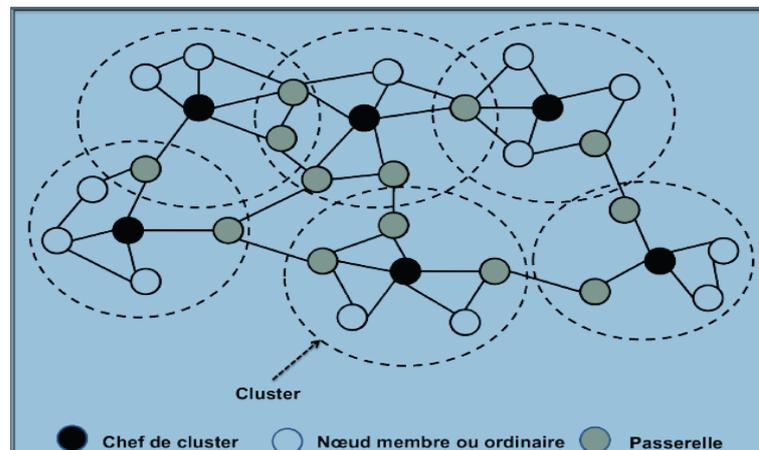


FIGURE 1.12: Exemple de structure de clusters

1.5.4 Mobilité des noeuds

Selon la mobilité des noeuds, on peut avoir deux grandes classes : les réseaux de capteurs statiques (static wireless sensor networks- WSNs) et les réseaux de capteurs mobiles (mobile wireless sensor networks- MWSNs). La plupart des domaines d'application existants utilisent des réseaux de capteurs statiques. Dans ce type de réseaux, tous les noeuds capteurs y compris la station de base sont statiques. Les réseaux de capteurs sans fil mobiles (RCSFMs) peuvent être simplement définis comme un réseau de capteurs sans fil (RCSF) dans lequel les noeuds capteurs sont mobiles. Les RCSFMs sont un petit domaine de recherche émergent comparé à son prédécesseur, les réseaux de capteurs statiques. RCSFMs sont beaucoup plus polyvalents

que les réseaux de capteurs statiques [4], car ils peuvent être déployés dans n'importe quel scénario et faire face aux changements rapides de la topologie. Le but de ce type de réseau est la plupart du temps l'exploration des zones inaccessibles ou dangereuses.

1.5.5 Capacité des noeuds dans le réseau

Dans cette classification, on distingue deux catégories : les réseaux de capteurs homogènes et les réseaux de capteurs hétérogènes [20]. Dans les réseaux de capteurs homogènes, tous les capteurs du réseau ont les mêmes caractéristiques. Ils ont les mêmes capacités du point de vue énergie, calcul et stockage. Tandis que dans les réseaux de capteurs hétérogènes, on trouve des capteurs sophistiqués qui ont plus de capacité en termes d'énergie et de traitement. Les capteurs sophistiqués peuvent être utilisés pour excuter les tâches les plus complexes comme les coordinateurs et les chefs des clusters,..., etc. De ce fait, l'utilisation de ces noeuds augmente la durée de vie du réseau. Cependant, il est difficile à mettre en oeuvre du fait qu'au moins chaque type de noeuds du réseau aura un code (programme) propre à lui. Ce qui augmente le coût du développement.

1.6 Caractéristiques des réseaux de capteurs sans fil

Les réseaux de capteurs sont caractérisés par :

– **Forte densité des noeuds :**

les RCSFs peuvent contenir des centaines voire des milliers de noeuds capteurs. La densité de noeud peut être aussi haute que $20 \text{ noeuds} = m^3$.

– **Ressources limitées :**

les capteurs sont des objets limités en termes de : capacité mémoire, puissance de traitement, bande passante et énergie.

– **Contrainte d'énergie :**

dans plusieurs applications, les noeuds capteurs sont déployés dans des environnements hostiles. Dans ce cas, la durée de vie du capteur dépend de la vie de sa batterie, ce qui exige la minimisation des dépenses énergétiques. De ce fait, l'énergie est une contrainte clé dans les réseaux de capteurs .

– **Déploiement de noeuds :**

il existe, plusieurs stratégies pour déployer un réseau de capteurs dans la zone à surveiller. Le choix d'une de ces stratégies dépend de l'objectif du réseau. Le déploiement peut être déterministe ou aléatoire. Dans la première stratégie, les capteurs sont placés manuellement et les données sont conduites via des chemins prédéterminés, mais cette solution devient impossible dès qu'on considère un très grand nombre de capteurs. En revanche, avec une approche aléatoire, les capteurs sont dispersés (exemple : lâchés d'un avion). Enfin, un mélange de ces deux techniques peut être envisagé.

– **Connectivité :**

la densité élevée de noeud dans les réseaux de capteurs exclue complètement leur isolement. Quoique, la connexion n'est pas toujours assurée à cause de la mobilité, le déploiement et les défaillances qui font varier la topologie du réseau .

– **Couverture limitée :**

dans les réseaux de capteurs, chaque noeud obtient une certaine vue locale de l'environnement, limitée par sa portée et sa précision. La couverture d'une vaste surface déterminée est composée de l'union de nombreuses couvertures de petite taille .

– **Auto organisation du réseau et la topologie dynamique :**

ceci peut être nécessaire, vue le grand nombre de noeuds de micro-capteurs et leur placements dans des endroits hostiles, où l'intervention humaine n'est pas faisable. D'ailleurs, les noeuds peuvent échouer (par manque d'énergie où destruction physique), comme de nouveaux noeuds peuvent rejoindre le réseau. Par conséquent, le réseau doit être capable de s'auto-organiser et de se modifier périodiquement de sorte qu'il puisse continuer à fonctionner .

– **Sécurité physique limitée :**

à l'instar des réseaux ad-hoc, les RCSFs sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques .

1.7 Applications des RCSFs

Les RCSFs [44] peuvent avoir beaucoup d'applications (voir la figure 1.13). Parmi elles, nous citons :

1. Découvertes de catastrophes naturelles : On peut créer un réseau autonome en dispersant les noeuds dans la nature. Des capteurs peuvent ainsi signaler des événements tels que feux de forêts, tempêtes ou inondations. Ceci permet une intervention beaucoup plus rapide et efficace des secours.
2. Détection d'intrusions : En plaçant, à différents points stratégiques, des capteurs, on peut ainsi prévenir des cambriolages ou des passages de gibier sur une voie de chemin de fer (par exemple) sans avoir à recourir à de coûteux dispositifs de surveillance vidéo.
3. Applications métier : On pourrait imaginer devoir stocker des denrées nécessitant un certain taux d'humidité et une certaine température (min ou max). Dans ces applications, le réseau doit pouvoir collecter ces différentes informations et alerter en temps réel si les seuils critiques sont dépassés.
4. Contrôle de la pollution : On pourrait disperser des capteurs au-dessus d'un emplacement industriel pour détecter et contrôler des fuites de gaz ou de produits chimiques. Ces applications permettraient de donner l'alerte en un temps record et de pouvoir suivre l'évolution de la catastrophe.
5. Agriculture : Des noeuds peuvent être incorporés dans la terre. On peut ensuite questionner le réseau de capteurs sur l'état du champ (déterminer par exemple les secteurs les plus secs afin de les arroser en priorité). On peut aussi imaginer équiper des troupeaux de bétail de capteurs pour connaître en tout temps, leur position ce qui éviterait aux éleveurs d'avoir recours à des chiens de berger.
6. Surveillance médicale : En implantant sous la peau de mini capteurs vidéo, on peut recevoir des images en temps réel d'une partie du corps sans aucune chirurgie pendant environ 24h. On peut ainsi surveiller la progression d'une maladie ou la reconstruction d'un muscle.
7. Contrôle d'édifices : On peut inclure sur les parois des barrages des capteurs qui permettent de calculer en temps réel la pression exercée. Il est donc possible de réguler le niveau d'eau si les limites sont atteintes. On peut aussi imaginer inclure des capteurs entre les sacs de sables formant une digue de fortune. La détection rapide d'infiltration

d'eau peut servir à renforcer le barrage en conséquence. Cette technique peut aussi être utilisée pour d'autres constructions tels que ponts, voies de chemins de fer, routes de montagnes, bâtiments et autres ouvrages d'art.

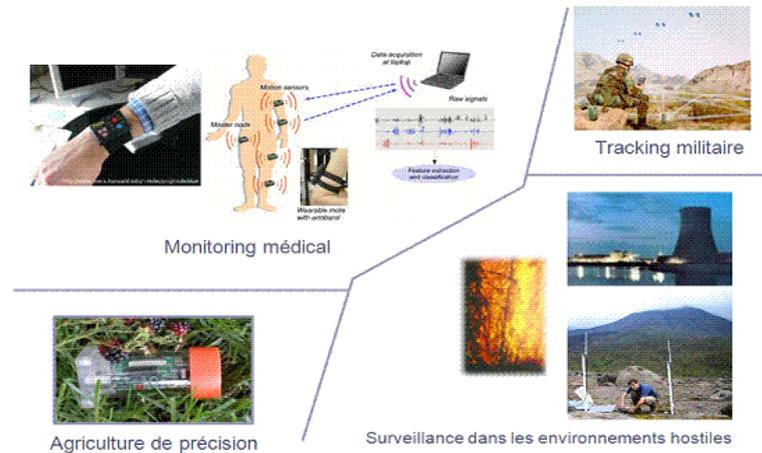


FIGURE 1.13: Applications des RCSF

1.8 Modèles de mobilité dans les RCSFs

1.8.1 La mobilité dans les RCSF

La mobilité est une caractéristique importante dans les réseaux de capteurs sans fil. Elle peut être considérée au niveau des noeuds capteurs destinés au captage ou au niveau des collecteurs, ou encore au niveau des puits. Lorsque la mobilité est trop rapide, la détection des voisins et la reconfiguration du réseau exigent habituellement un nombre important de messages de contrôle de la topologie et donc une consommation d'énergie importante.

1.8.2 Architecture du RCSF mobile

L'architecture du RCSF mobile est différente de celle d'un RCSF statique, le puits continue à se déplacer autour ou à l'intérieur du champ de détection pour la collecte efficace de données. Une référence de l'architecture du RCSF mobile est représentée sur la figure 1.14.

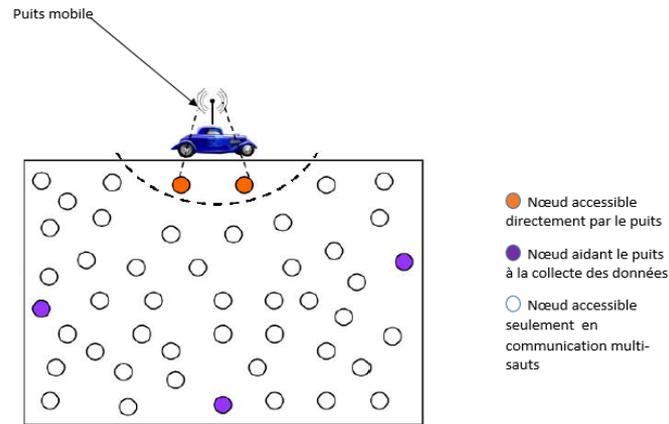


FIGURE 1.14: Architecture du RCSFs mobile

Les principaux composants d'un RCSF mobile sont donnés comme suit :

Noeuds réguliers : sont les noeuds capteurs ordinaires qui sont déployés dans le champ de capteur pour détecter un phénomène d'intérêt. Lors de la détection des événements, ces noeuds diffusent leurs données de manière coopérative vers un puits mobile. En fonction de leur emplacement dans le champ de capture, les noeuds pourraient fonctionner comme des relais transmettant ainsi d'autres données vers un puits mobile.

Puits mobile : Selon le scénario d'application, ça pourrait être un ou plusieurs puits mobiles qui se déplacent autour ou à l'intérieur du champ de capteur pour la collecte de données. De tels dispositifs sont considérés comme des dispositifs sans contrainte en termes de leurs ressources. Le puits mobile peut être un noeud capteur fixé à un être humain, voiture, animal ou un robot.

Assistants du puits (Facultatif) : Dans certaines applications, des noeuds spéciaux sont déployés dans des positions stratégiques qui fournissent une assistance au puits dans la collecte des données. Ces appareils sont également considérés riches en énergie. Dans le cas où ces assistants sont statiques, ils deviennent des collecteurs de données intermédiaires pour les noeuds capteurs et puis transmettent les données collectées au puits mobile dès son arrivée. Dans le cas de la mobilité, ils sont destinés à assurer une couverture près du champ de capture pour les services de communication en temps réel dans certaines applications [25].

1.8.3 Pourquoi la mobilité du puits ?

Le puits a pour principal objectif la collecte des différentes données générées par les capteurs. Plusieurs travaux considèrent que ce puits est statique ou immobile. Cependant, cette hypothèse n'est pas toujours réalisable notamment pour des raisons de sécurité ou de déploiement. De plus, la mobilité du puits permet d'augmenter la durée de vie du réseau en évitant de surcharger les noeuds présents dans le voisinage du puits [14]. Plusieurs raisons expliquent l'intérêt de l'utilisation des puits mobiles, par rapport aux puits statiques, notamment pour l'amélioration des performances réseau

1.8.4 Les avantages de la mobilité du puits

Dans presque toutes les applications des RCSFs, le puits est considéré comme une entité sans contrainte en termes de ressources (réserve d'énergie, de puissance de traitement, capacité de communication, etc.). De même, dans plusieurs applications des réseaux de capteurs, la mobilité des puits peut être réalisée par la fixation d'un dispositif récepteur vers une entité mobile comme humain, animal, robot, ou d'un véhicule qui peut se déplacer à l'intérieur du champ de détection pour la collection des données. Des économies d'énergie considérables peuvent être obtenues par le déploiement d'un puits mobile dans le champ de captage KINALIS et AL [5] ont identifié plusieurs avantages potentiels de la mobilité du puits qui sont décrits comme suit :

- Augmentation de la durée de vie du capteur :

En exploitant la mobilité du puits, non seulement le problème d'énergie qui est réduit, mais ceci améliore également la durée de vie des noeuds. Les noeuds capteurs sont considérés comme des dispositifs à contrainte d'énergie alors que le puits, étant externe au réseau, n'a pas de contraintes énergétiques. étant donné que le module de communication est considéré comme le principal consommateur des réserves d'énergie d'un noeud, si le puits se rapproche de la source d'évènement des noeuds, de plus grandes économies d'énergie pourraient être obtenus, ce qui limite la communication multi-saut.

- Amélioration de la couverture :

Un puits mobile peut potentiellement couvrir les réseaux clairsemés en raison de sa fonction de mobilité. De même, le puits mobile peut également passer par les zones problématiques où il y a une obstruction dans le chemin de propagation, telle que de

gros rochers.

- Amélioration du débit et la fiabilité des données :
En exploitant la mobilité du puits, on peut également obtenir un meilleur débit et des données fiables. Si le puits se déplace vers la zone d'intérêt, il ne va pas seulement réduire le nombre de transmissions mais aussi réduire la probabilité d'erreurs de transmission et les risques de collisions. Minimiser les retransmissions améliore non seulement le débit du réseau, mais prolonge également sa durée de vie.
- Amélioration de la sécurité :
En exploitant la mobilité du puits, il est relativement difficile d'écouter les informations sur le réseau, de ce fait, le réseau est moins exposé aux menaces de sécurité . En outre, un adversaire peut obtenir les informations concernant uniquement une petite zone en raison du nombre réduit de transmissions multi-sauts. De même, les attaques possibles ciblant les noeuds capteurs de passerelle pour perturber le fonctionnement du réseau ne sont pas possibles en raison de la mobilité du puits, il n'y a pas de tels noeuds géostratégiques par lesquels les messages doivent toujours passer.

1.8.5 Les enjeux liés à la mobilité

La mobilité du puits soulève plusieurs défis lors de la conception et la mise en place d'un protocole de routage. N'importe quel protocole de routage conçu pour un RCSFs à puits mobile doit effectuer les opérations suivantes [31] :

- Informer les voisins sur la présence où la rupture du lien avec le puits mobile.
- Propager des mises à jour topologiques du puits pour assurer la connectivité.
- Réduire les risques de perte de paquets tandis que le puits se déplace d'un point à un autre.

Cependant, ces opérations ne peuvent pas être prises d'une manière holistique car cela augmente considérablement la consommation d'énergie du noeud capteur pour chaque mouvement du puits mobile. Les éléments suivants sont les différents défis qui se posent en raison de la mobilité du puits [25] :

– **Détection de la présence du puits :**

D'abord la présence du puits doit être détectée par les noeuds capteurs qui sont

dans sa portée de communication. La détection de la présence du puits est très influencée par la vitesse du puits mobile et les périodes d'activités des noeuds, puisque un noeud en mode veille ne sera pas en mesure de détecter la présence d'un puits mobile. De même la forte mobilité du puits introduit un court temps de contact avec les noeuds capteurs qui provoque par conséquent un taux de perte de paquets élevé .

– **La gestion de la période d'activité en fonction de la mobilité :**

Pour propager les mises à jour topologiques d'un puits, les noeuds capteurs ont besoin d'être en mode écoute. Si le mouvement du puits peut être prédit ou calculé en exploitant les connaissances sur le modèle de la mobilité du puits, il pourrait aider les noeuds à optimiser la détection du puits. Dans les situations où les heures de visite sont connues à priori ou calculables avec certaine précision, les cycles de service des noeuds peuvent être ajustés en conséquence pour les mettre en mode actif à l'heure d'arrivée prévue du puits mobile. Cependant, ceci est seulement applicable dans des situations où le puits suit toujours une certaine trajectoire tout en conservant une vitesse constante.

– **Augmentation de la latence :**

Il y a toujours un compromis entre la consommation d'énergie et la latence dans un RCSF. La latence dépend directement du modèle de mobilité du puits (vitesse, direction, et l'intervalle de pause). Si les dernières informations de localisation d'un puits mobile ainsi que les informations sur la mobilité du puits sont rapidement propagées dans l'ensemble du réseau, le délai de bout en bout de latence peut être réduit en permettant aux noeuds d'ajuster leurs routes vers le puits. Cependant, cela provoquerait une énorme consommation d'énergie et diminuerait ainsi la durée de vie du réseau. D'autre part, si la dernière information de localisation se propage peu fréquemment ou seulement pour un nombre limité de noeuds, le chemin adopté pour la livraison des données peut ne pas être optimal en termes de nombre de sauts. De plus, la latence est fortement influencée par la période de pause du puits mobile. Dans certaines situations, le puits se dirige vers chaque noeud pour collecter les données. Dans de tels scénarios, si la taille des données est assez grande qu'elles ne peuvent être téléchargées par le puits durant le temps de contact, les noeuds devront attendre la prochaine tournée du puits mobile. Mais, cela produirait une augmentation de latence.

– **Augmentation de taux de perte de paquets :**

En raison de l'indisponibilité des contacts fixes avec le puits mobile, les données sont transmises par chaque noeud capteur vers la dernière position connue du puits mobile. Cependant, si la dernière information de mobilité ne se propage pas dans l'ensemble du réseau, la transmission des messages serait compromise. Ce problème se complique d'avantage quand un puits mobile aurait quitté sa dernière position de manière significative, entraînant finalement la perte des données en raison d'un temps de parcours long.

1.8.6 Les modèles de mobilité du puits dans les réseaux de capteurs

Il existe trois modèles de mobilité de base d'un puits mobile dans un RCSF [25] : la mobilité aléatoire, prévisible et contrôlée. La mobilité aléatoire et prévisible des puits ne sont pas sous le contrôle du gestionnaire du réseau.

La mobilité aléatoire

Le mouvement du puits est fait d'une manière aléatoire en ce qui concerne la direction et la vitesse du mouvement. Dans ce type de mobilité, le puits est autonome dans son mouvement. La prochaine position du puits est imprévisible. Pour les applications tolérantes au délai, la mobilité aléatoire du puits prolonge la durée de vie du réseau [15] et elle est particulièrement applicable dans les situations où le déploiement du noeud est inconnu. Elle est utilisée quand le dispositif est attaché à une unité mobile tel un animal dont on ne peut prévoir son mouvement.

La mobilité prévisible

La mobilité prévisible est la plus simple parmi les trois modèles de mobilité. Elle est adaptée par le puits s'il suit toujours une certaine trajectoire par exemple une ligne droite, la périphérie de captage du champ. En utilisant ce modèle de mobilité, les noeuds peuvent prévoir l'heure de la visite du puits mobile [36] et donc d'optimiser la détection et la livraison de données. Ce type de modèle de mobilité est particulièrement attrayant pour des applications telles que les parkings ou le puits mobile (pilote de voiture) peut se renseigner sur la disponibilité des places de stationnement dans une zone d'intérêt.

La mobilité contrôlée

La mobilité contrôlée se réfère à la situation lorsque l'observateur d'un réseau de capteurs peut contrôler le mouvement du puits mobile [23]. La mobilité contrôlée est utilisée par des

puits si leur dispositif est relié à une unité mobile tel qu'un robot. La mobilité des puits contrôlée est basée sur l'hypothèse que les unités mobiles sensibles au contexte sont à la disposition du réseau. Le dispositif sensible au contexte calcule les éventuelles positions futures sur la base des données générées par le champ de captage [15]. Selon l'objectif de l'application, le puits mobile adapte son mouvement (à la fois la vitesse et la trajectoire) d'une manière déterministe pour obtenir de meilleurs résultats. Par exemple, si l'objectif est de prolonger la durée de vie du réseau, le puits fait son prochain déplacement vers les zones riches en énergie pour une consommation équilibrée d'énergie des noeuds. De même, dans les applications où la latence est critique, le puits se déplace vers la zone d'intérêt pour réduire les communications multi-saut. Cependant, le planning des visites du puits mobile doit être soigneusement adopté pour éviter des visites rares pour des segments particuliers du réseau qui peuvent entraîner une grande latence [38]. En termes de coût, la mobilité contrôlée n'est pas le meilleur choix, mais si le but est de réduire la latence, la mobilité contrôlée donne de meilleurs résultats que la mobilité aléatoire et prévisible [27].

1.9 Position de problème

Les RCSFs se sont développés à un point où l'électronique ne peut pas pour l'instant apporter une solution à tous les problèmes surtout au problème de batterie qui ont une durée de vie limitée. En effet, les ressources énergétiques de ces derniers sont non renouvelable, en plus vu leur déploiement dans des endroits hostiles où il n'est pas possible de les atteindre.

Notre problème s'inscrit dans le cadre de la minimisation de l'énergie d'un RCSFs en considérons que la station de base (SB) est mobile. Il s'agit de déterminer l'algorithme et d'appliquer les techniques de la recherche opérationnelle afin d'atteindre cet objectif, qui nous permettra de trouver un chemin optimal de la station de base mobile pour visiter tous les chefs de cluster une et une seule fois et détecter les informations, les questions pouvant être posées sont :

- Quels sont les méthodes d'optimisation utilisées pour assurer la minimisation de la consommation d'énergie dans un RCSF à une station de base mobile ?
- Comment le clustering des noeuds peut assurer une consommation optimale d'énergie ?
- Comment peut on trouver une tournée de la station de base mobile ?
- Quels sont les algorithmes utilisés pour optimiser le chemin de routage de la station de base ?

Pour répondre à ces questions nous allons modéliser ce problème comme étant un problème d'optimisation combinatoire qui fait l'objet du deuxième chapitre.

1.10 Conclusion

Les réseaux de capteurs sans fil présentent un intérêt considérable et une nouvelle étape dans l'évolution des technologies de l'information et de la communication. Cette nouvelle technologie suscite un intérêt croissant vu la diversité de ces applications : santé, environnement, . . . etc.

Dans ce premier chapitre, nous avons présenté les réseaux de capteurs sans fil, leurs architectures de communication, leurs caractéristiques, ainsi que les domaines d'applications de ce genre de réseau et les modèles de mobilité existants dans la littérature.

CHAPITRE 2

PROBLÈME DE TOURNÉES

2.1 Introduction

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle. Elle se trouve au carrefour de la théorie des graphes, de la programmation linéaire et de la programmation en nombres entiers. Son importance se justifie d'une part, par la grande difficulté des problèmes d'optimisation et d'autre part, par les nombreuses applications pratiques pouvant être formulées sous forme d'un problème d'optimisation combinatoire.

2.2 Rappel sur l'optimisation combinatoire

2.2.1 Définition d'un problème d'optimisation combinatoire

En mathématique, un problème d'optimisation consiste à chercher une instanciation d'un ensemble de variables qui peuvent être soumises (ou pas) à des contraintes, de façon à maximiser ou minimiser un ou plusieurs critères. Lorsque le domaine de valeurs des variables est discret, on parle alors de problème d'optimisation combinatoire (POC). Formellement un problème d'optimisation combinatoire est défini comme suit :

$$\min_{s \in S} f(s), \tag{2.1}$$

où, S est l'ensemble des solutions réalisables ou admissibles du problème(2.1).

La fonction $f : S \rightarrow R$ est dite fonction objectif (critère) à optimiser.

Le problème de minimisation (2.1) se ramène facilement à un problème de maximisation, telle que :

$$\max_{s \in S} f(s) = - \min_{s \in S} (-f(s)). \quad (2.2)$$

Pour un problème de minimisation, il s'agit de trouver une solution $s \in S$ réalisable, qui minimise la fonction f .

Une solution $s \in S$ pour le problème (2.1) peut être un minimum local ou global, définis comme suit :

- **Minimum local** : Une solution $s \in S$ est un minimum local du problème (2.1), s'il existe un voisinage $N(s)$, vérifiant, $f(s) \leq f(s'), \forall s' \in N(s)$,

où, $N(s) = \{x \in S : \|s - x\| < \varepsilon\}$, avec $\varepsilon > 0$ un nombre réel.

- **Minimum global** : Une solution $s \in S$ est un minimum global pour (2.1) s'il vérifie :

$$f(s) \leq f(s'), \forall s' \in S.$$

2.2.2 Problèmes classiques d'optimisation combinatoire

Le problème du plus court chemin

Soit un graphe $G = (X, V)$, où X est l'ensemble des sommets et V est l'ensemble des arcs.

On appelle le problème du plus court chemin le problème suivant :

Etant donné un graphe G , nous associons à chaque arc u un nombre $l(u) \geq 0$ que nous appellerons la longueur de l'arc u . Trouver un chemin élémentaire μ , allant d'un sommet source s à un sommet puits t , tel que la longueur totale :

$$l(\mu) = \sum_{u \in \mu} l(u)$$

soit aussi petite que possible.

Le problème de sac à dos

Le problème de sac à dos modélise des situations analogues au remplissage d'un sac à dos, i.e. étant donné un sac de capacité b et n objets, où chaque objet j possède un poids P_j et une valeur v_j . Le problème pose est le suivant : comment remplir le sac de sorte que le poids

total des objets choisis n'excède pas la capacité b du sac tout en maximisant la valeur total. Le problème du sac à dos est modélisé par le PL à variables bornées suivantes :

$$\left\{ \begin{array}{l} \max Z = \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n P_j x_j \leq b \\ x_j \in \{0, 1\}, j = 1, 2, \dots, n. \end{array} \right. \quad (2.3)$$

où n est le nombre d'objets et les variables de décisions $x_j, j = 1, 2, \dots, n$ sont définies comme suit :

$$x_j = \begin{cases} 1, & \text{Si l'objet } j \text{ est sélectionné,} \\ 0, & \text{Sinon.} \end{cases} \quad (2.4)$$

Le problème du voyageur de commerce

Le problème du voyageur de commerce (PVC) est l'un des problèmes classiques d'optimisation combinatoire. Il s'agit d'un voyageur qui souhaite visiter n villes données en passant par chaque ville exactement une et une seule fois (établir une tournée). Le problème consiste à trouver une tournée (un circuit hamiltonien) de longueur minimale, et ce en démarrant d'une ville donnée et passer par toutes les villes une et une seule fois et revenir au point de départ. La notion de distance peut être remplacée par d'autres notions, telles que le temps de parcourt ou le coût.

Considérons la distance entre les villes i et j est d_{ij} :

1. Les variables de décision :

$$x_{ij} = \begin{cases} 1, & \text{Si la ville } j \text{ suit immédiatement la ville } i \text{ dans le parcours,} \\ 0, & \text{Sinon.} \end{cases} \quad (2.5)$$

2. Les contraintes :

(a) Le nombre de villes que le voyageur visite après la ville i est 1.

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1 \dots n. \quad (2.6)$$

(b) Le nombre de villes que le voyageur visite avant la ville j est 1.

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1 \dots n. \quad (2.7)$$

(c) La contrainte de sous-tournée : Soit S un sous ensemble de villes visitées.

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad \emptyset \neq S \subset \{1, 2, \dots, n\}. \quad (2.8)$$

3. **La fonction objectif** : minimiser la longueur parcourue du voyageur.

$$\min Z = \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \quad (2.9)$$

4. **Le modèle mathématique** :

$$\left\{ \begin{array}{l} \min Z = \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \\ \sum_{j=1}^n x_{ij} = 1, \quad i = 1 \dots n \\ \sum_{i=1}^n x_{ij} = 1, \quad j = 1 \dots n \\ \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \emptyset \neq S \subset \{1, 2, \dots, n\} \\ x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n \end{array} \right. \quad (2.10)$$

2.2.3 Méthodes de résolution pour le PVC

Il existe deux grandes catégories de méthodes de résolution : les méthodes exactes et les méthodes approchées. Les méthodes exactes permettent d'obtenir une solution optimale, mais le temps de calcul peut être long si le problème à résoudre est de grande de taille. Les méthodes approchées englobent des heuristiques et des métha-heuristique. Une heuristique est un algorithme spécifique pour un problème donné et permet d'obtenir en un temps raisonnable une seule solution (un minimum local). Les métha-heuristiques sont des méthodes d'optimisation globale permettant de trouver une solution (ou une population de solutions) proche de l'optimum.

Méthodes exactes :

- **Branch and Bound** :

Pour le problème du voyageur de commerce, l'une des méthodes exactes les plus classiques

et les plus performantes est l'algorithme de séparation et évaluation, plus connu sous son appellation anglaise Branch and Bound (B&B) [18]. C'est une approche arborescente de recherche d'une solution optimale par séparations et évaluations, en représentant les états solutions par un arbre d'états, avec des noeuds, et des feuilles. Le branch-and-bound est basé sur trois principes :

- L'évaluation,
- La séparation,
- La stratégie de parcours.

– **L'évaluation :**

L'évaluation permet de réduire l'espace de recherche en éliminant quelques sous ensembles qui ne contiennent pas la solution optimale. L'objectif est d'essayer d'évaluer l'intérêt de l'exploration d'un sous-ensemble de l'arborescence. Le branch-and-bound utilise une élimination de branches dans l'arborescence de recherche de la manière suivante : la recherche d'une solution de coût minimal, consiste à mémoriser la solution de plus bas coût rencontré pendant l'exploration, et à comparer le coût de chaque noeud parcouru à celui de la meilleure solution. Si le coût du noeud considéré est supérieur au meilleur coût, on arrête l'exploration de la branche et toutes les solutions de cette branche seront nécessairement de coût plus élevé que la meilleure solution déjà trouvée.

– **La séparation :**

La séparation consiste à diviser le problème en sous-problèmes. Ainsi, en résolvant tous les sous-problèmes et en gardant la meilleure solution trouvée, on est assuré d'avoir résolu le problème initial. Cela revient à construire un arbre permettant d'énumérer toutes les solutions. L'ensemble de noeuds de l'arbre qu'il reste encore à parcourir comme étant susceptibles de contenir une solution optimale, c'est-à-dire encore à diviser, est appelé ensemble des noeuds actifs.

– **La stratégie de parcours :**

La largeur d'abord : Cette stratégie favorise les sommets les plus proches de la racine en faisant moins de séparations du problème initial. Elle est moins efficace que les deux autres stratégies présentées.

La profondeur d'abord : Cette stratégie avantage les sommets les plus éloignés de la racine (de profondeur la plus élevée) en appliquant plus de séparations au problème initial. Cette voie mène rapidement à une solution optimale en économisant la mémoire.

Le meilleur d'abord : Cette stratégie consiste à explorer des sous problèmes

possédant la meilleure borne. Elle permet aussi d'éviter l'exploration de tous les sous-problèmes qui possèdent une mauvaise évaluation par rapport à la valeur optimale.

Méthodes heuristiques :

- **Méthode des plus proche voisin :**

L'algorithme de plus proche voisin (PPV) [39] est une heuristique simple et très rapide, mais la solution calculée n'est pas forcément optimale. Il est souvent utilisé pour constituer rapidement un trajet acceptable, sa complexité est de l'ordre de $o(n^2)$.

Les étapes de cette algorithme sont :

Algorithm 1: PPV.

```

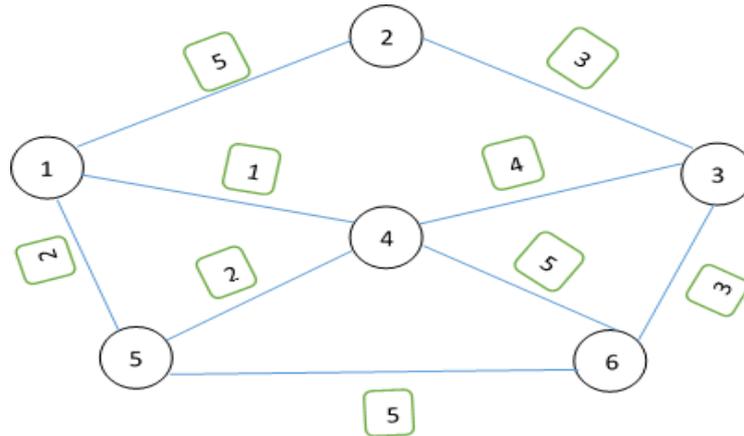
1 Choisir un sommet source  $s \in S$ ;
2 La ville courante  $c = s$ ;
3 while  $\exists x \in S$  non visité do
4   |   Sélectionné  $v$ , le voisin le plus proche de  $c$ , tq  $\forall v, v' \in S, d_{cv} \leq d_{cv'}$ ;
5   |   Aller de  $c$  vers  $v$ ;
6   |    $c = v$ ;
7 end
8  $c = s$ ;

```

Exemple d'application :

Soit $G = (V, A)$ un graphe composé d'un ensemble V de $n = 6$ villes, et d'un ensemble A d'arêtes. Soit d_{ij} une matrice de distances relative à A . On cherche à trouver un circuit Hamiltonien, c'est à-dire un circuit qui ne passe qu'une seule fois par chaque ville et de longueur minimal.

Ci-dessous un petit problème d'un voyageur de commerce qu'on va résoudre avec l'approche heuristique PPV :



La matrice des distances entre les 6 villes est la suivante :

$$D = \begin{bmatrix} 0 & 5 & 0 & 1 & 2 & 0 \\ 5 & 0 & 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 4 & 0 & 3 \\ 1 & 0 & 4 & 0 & 2 & 5 \\ 2 & 0 & 0 & 2 & 0 & 5 \\ 0 & 0 & 3 & 5 & 5 & 0 \end{bmatrix}$$

Initialisation : On choisit $V_0 = 1$, comme ville de départ.

Itération 1 : Sélectionner la prochaine ville non visitée V_1 la plus proche de $V_0 = 1$, tel que :

$$\min\{d_{12}, d_{14}, d_{15}\} = \min\{5, 1, 2\} = d_{14} = 1 \Rightarrow V_1 = 4.$$

Itération 2 : Sélectionner la prochaine ville non visitée V_2 la plus proche de $V_1 = 4$ vérifiant :

$$\min\{d_{43}, d_{45}, d_{46}\} = \min\{4, 2, 5\} = d_{45} = 2 \Rightarrow V_2 = 5.$$

Itération 3 : Sélectionner la prochaine ville non visitée V_3 la plus proche de $V_2 = 5$ tel que :

$$\min\{d_{56}\} = 5 \Rightarrow V_3 = 6.$$

Itération 4 : Sélectionner la prochaine ville non visitée V_4 la plus proche de $V_3 = 6$ tel que :

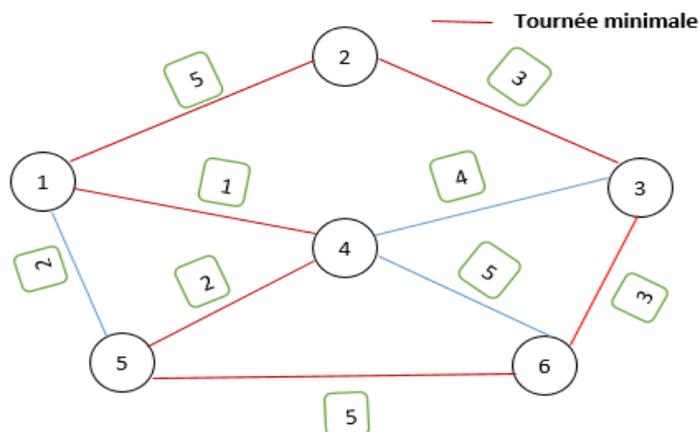
$$\min\{d_{63}\} = 3 \Rightarrow V_4 = 3.$$

Itération 5 : Sélectionner la prochaine ville non visitée V_5 la plus proche de $V_4 = 3$ tel que :

$$\min\{d_{32}\} = 3 \Rightarrow V_5 = 2.$$

Itération 6 : Lorsque toutes les villes sont sélectionnées le voyageur revient au point de départ $V_0 = 1$.

La tournée obtenue par la méthode est illustrée dans la figure suivante : ●



Méthode des 2 optimalité :

En optimisation, 2-opt est un algorithme de recherche locale proposé par Georges A. Croes en 1958 pour résoudre le problème du voyageur de commerce en améliorant une solution initiale.

2-opt est un algorithme itératif de recherche locale plutôt simple, et qui consiste à parcourir le tour courant à la recherche de deux arêtes $(a; b)$ et $(c; d)$ qui peuvent être remplacées par les arêtes $(a; c)$ et $(b; d)$ ou $(a; d)$ et $(b; c)$ pour

former un nouveau tour plus court. Si le nouveau tour ne l'est pas, alors nous gardons la configuration précédente et nous recherchons une autre paire d'arêtes.

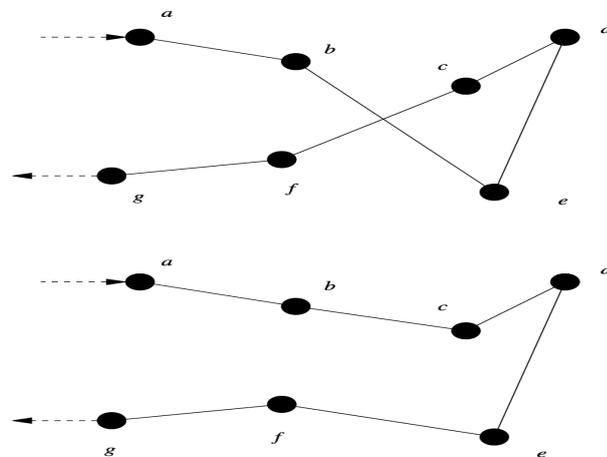


FIGURE 2.1: Exemple de permutation 2-opt [42]

Il faut noter que lorsque l'on inverse l'ordre de parcours de deux villes, il faut également inverser l'ordre de parcours de toutes les villes qui se trouvent entre elles. Et il s'agit là d'une opération qui peut coûter cher à la longue. Une des améliorations possibles dans ce cas est d'inverser la partie qui contient le moins de villes. Il existe également d'autres techniques qui permettent d'améliorer la complexité effective de l'heuristique, comme par exemple le calcul d'une liste des k voisins les plus proches pour chaque sommet, afin de réduire le nombre d'arêtes candidates avec lesquelles comparer chaque arête.

Métaheuristiques :

Contrairement aux heuristiques qui construisent une ou plusieurs solutions en partant d'une solution vide, les métaheuristiques débutent par une solution initiale arbitraire $s_0 \in X$ sur l'ensemble des solutions réalisables et tentent de rendre meilleure celle-ci. Ces méthodes sont prisées pour leur efficacité et leur existence en termes d'adaptation à n'importe quel problème d'optimisation. Le point fort de ces méthodes est leur aptitude à échapper aux minima (respectivement

maxima) **locaux**, en acceptant par fois durant l'exécution de leur programmes, des solutions qui dégradent temporairement la fonction objectif afin d'explorer toute solution potentielle qui pourrait être un minimum (respectivement maximum) **global**.

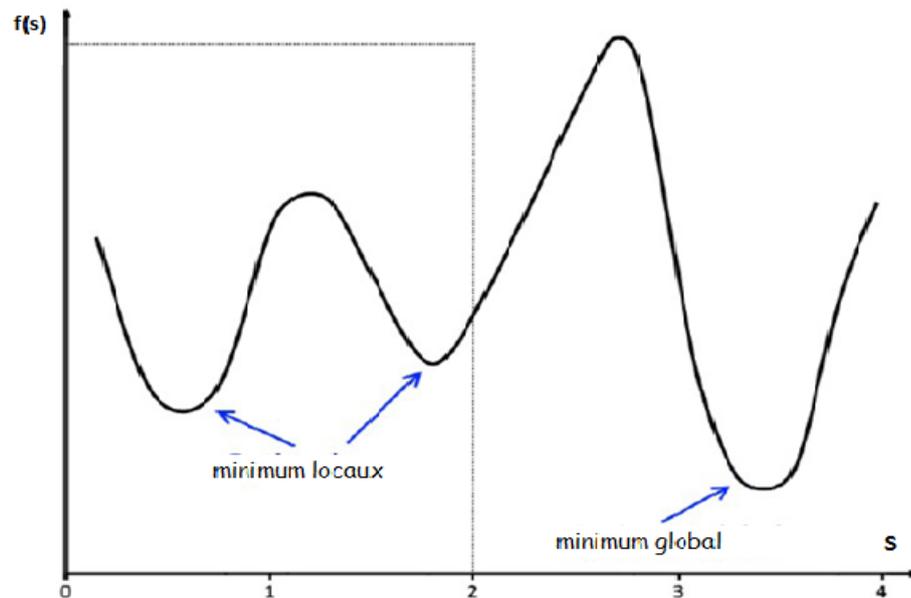


FIGURE 2.2: Minima locaux et globaux d'une fonction

Un grand nombre de métaheuristiques existe, nous présentons quelques-unes dans la suite de ce chapitre, nous les avons classés par type : celles à solution unique et celles à population de solutions :

- **Méthodes de recherche locale (à solution unique) :**

Ces méthodes partent d'une solution initiale (obtenue par tirage aléatoire, ou provient d'une autre méthode approchée) et la déplace de façon itérative vers une solution voisine. Cette méthode est applicable seulement si, une notion de voisinage est définie.

Habituellement, chaque solution a plus d'une solution voisine ; le choix de celle vers laquelle se déplacer est pris en utilisant seulement les informations sur les solutions voisines de la solution courante, d'où le terme de recherche locale. Ces

méthodes ne manipulent qu'une seule solution à la fois [7, 24].

Nous présentons un algorithme à solution unique :

Le recuit simulé

Le recuit simulé est une métaheuristique inspirée d'un mécanisme naturel en métallurgie, celui de l'alliage d'un élément métallique avec d'autres éléments chimiques. Il se base sur les travaux faits par *Metropolis en 1953*, qui décrivent le processus de recuit physique consistant à chauffer un matériau à une température suffisante pour que les atomes retrouvent la liberté du déplacement afin d'aboutir à un équilibre physico-chimique. Cette opération est suivie d'un refroidissement lent où les atomes cherchent à former de nouvelles liaisons, pour obtenir une structure régulière du matériau à l'état solide.

Un état donné de la matière représente dans la méthode du recuit simulé, une solution s du problème à résoudre, l'énergie du système représente la fonction objectif f à optimiser, donc un état d'équilibre où l'énergie $f(s)$ est optimale représente la solution optimale du problème [10, 13] .

Les étapes de l'algorithme du recuit simulé sont représentées comme suit :

Algorithme 1.1 - Recuit Simulé [13, 17]

1. Générer aléatoirement une configuration initiale $s = s_0$ dont correspond l'énergie initiale $E = E_0$,
2. Initialiser la température $T = T_0$ en fonction du schéma de refroidissement,
3. Générer d'une manière aléatoire une configuration voisine s' de la configuration actuelle s , en déplaçant au hasard un atome quelconque,
4. Calculer la variation de l'énergie $\Delta E = f(s') - f(s)$
Si $\Delta E < 0$, alors la nouvelle solution améliore la fonction objectif et diminue l'énergie, donc elle est acceptée.
Sinon, elle sera acceptée avec une probabilité égale à $e^{-\frac{\Delta E}{T}}$.
5. Répéter 3. et 4. jusqu'à ce que la configuration optimale soit atteinte.

6. Décroître la température et répéter jusqu'à ce que le système se solidifie.

• **Méthodes évolutives (à population de solutions) :**

Les Algorithmes Génétiques (évolutionnaires)

Les algorithmes génétiques développés en 1975 par *Holland*, ont été efficacement utilisés pour résoudre plusieurs problèmes d'optimisation en imitant le processus d'évolution naturelle des espèces dans un environnement donné, suivant le modèle Darwinien selon lequel, les individus de chaque espèce les mieux adaptés à leur environnement, ont plus de chance de survivre et de se reproduire au fil des générations par la transmission des meilleures caractéristiques qui ont permis leurs survie, donnant ainsi des descendants de plus en plus adaptés à leurs environnement.

Les algorithmes génétiques sont des méthodes qui font évoluer un ensemble de solution appelé *population* qui représente l'espace de solution d'un problème donné, chaque point de cet espace est représenté par *un chromosome* appelé aussi *individu*. L'ensemble des variables X qui forment un chromosome (solution) sont appelés *gènes*. A chaque individu est associée une valeur dite *fitness* qui mesure son adaptation à son environnement (à l'objectif visé).

La particularité de ces algorithmes est qu'ils utilisent un codage des paramètres et non pas les paramètres eux même.

Un algorithme génétique commence par générer une population aléatoire P de N individus qui évoluent sur plusieurs générations, dans cette évolution les générations successives des différentes populations préservent une taille constante de N individus.

Le processus d'évolution d'un algorithme génétique est basé sur trois opérateurs principaux : la sélection, le croisement et la mutation [10, 12, 26].

Nous présentons ces opérateurs sous l'hypothèse que le codage des paramètres est binaire où les individus sont représentés sous forme de chaînes de bits (qui

peuvent prendre les valeurs 0 ou 1) comme l'illustre la figure suivante :

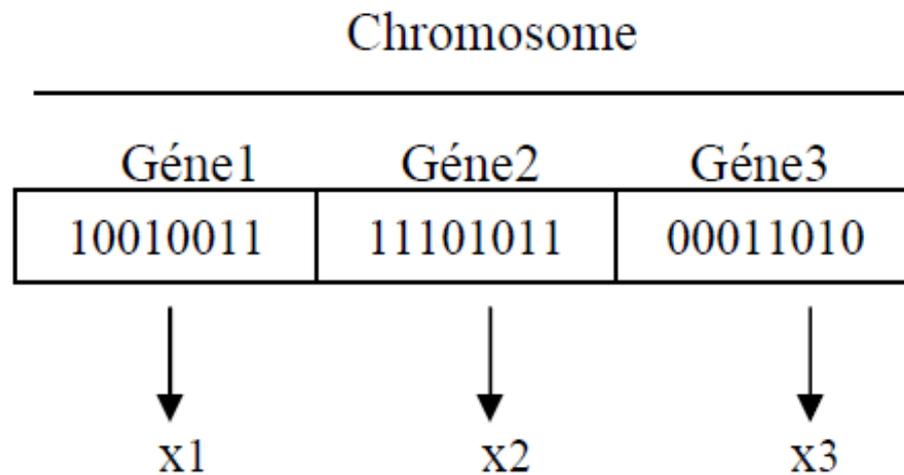


FIGURE 2.3: Illustration du codage des variables d'optimisation

Les opérateurs génétiques :

– Opérateur de sélection

La sélection est un processus qui consiste à choisir parmi tous les individus de la population ceux qui vont participer à la construction d'une nouvelle génération où ce choix est basé essentiellement sur les valeurs d'adaptation de chaque individu et du critère à optimiser.

Cet opérateur intervient dans deux étapes. Premièrement à l'étape de la sélection des individus parents qui vont se reproduire pour construire de nouveaux individus enfants. Ensuite, il est appliqué à la fin de chaque itération de l'algorithme pour sélectionner les individus qui vont survivre et construire la nouvelle population [19].

Il existe plusieurs techniques de sélection, les principales utilisées sont :

- *La sélection par classement* : elle consiste à classer dans un ordre croissant (ou décroissant selon l'objectif) les valeurs des fonctions d'adaptation et à retenir un nombre fixé d'individus. Seuls les individus les plus forts sont conservés.
- *La sélection proportionnelle* ou *sélection par roue de loterie* : qui associe à chaque individu un secteur de la roulette proportionnelle à la valeur de sa fonction d'adaptation. Ainsi, même les individus les plus faibles ont une chance de survivre.

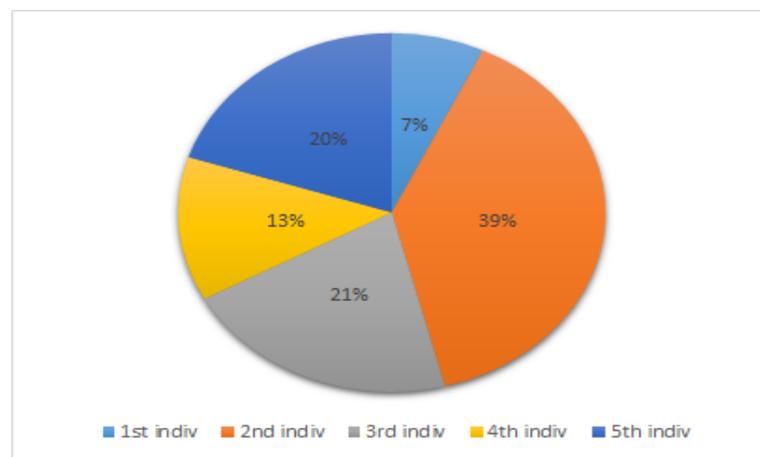


FIGURE 2.4: Sélection par roulette

– L'opérateur croisement

Le croisement est appliqué avec une probabilité P_c à deux individus parents préalablement sélectionnés, qui échangent des parties de leurs chaînes, pour former deux nouveaux individus enfants [19].

Usuellement, le croisement se fait selon deux modes :

- *Le croisement en un point* : consiste à choisir aléatoirement un point de coupure identique pour les chromosomes des deux parents, par la suite les sous chaînes situées après ce point sont inter-changées pour former les deux enfants.

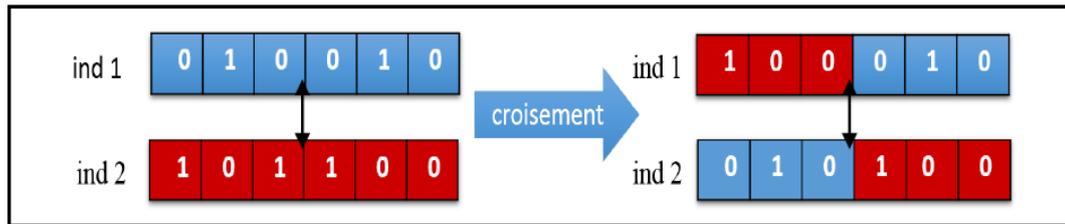


FIGURE 2.5: Le croisement en un point

- *Le croisement en deux points* : Dans ce type de croisement, deux points de coupure sont choisis au hasard et le contenu entre ces points est interchangé pour former les chaînes des descendants. Cet opérateur est généralement considéré comme plus efficace que le précédent.

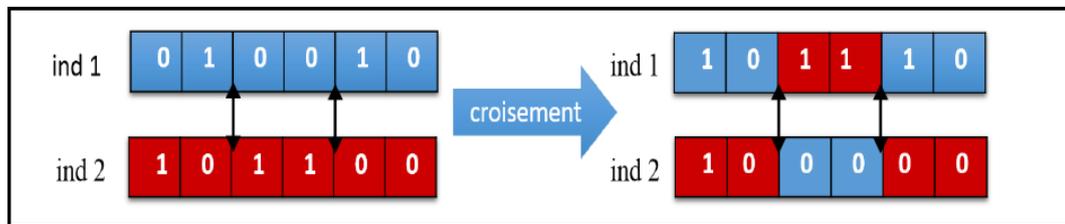


FIGURE 2.6: Le croisement en deux points

– L'opérateurs de mutation

Selon une probabilité P_m , les descendants sont mutés, en modifiant aléatoirement la valeur d'un ou plusieurs composants de leur chaînes. Il ne crée pas forcément de meilleures solutions au problème, mais garantit la diversité de la population et évitent l'établissement de populations uniformes incapables d'évoluer [22].

La mutation classique consiste à remplacer dans un chromosome binaire un bit par son inverse.

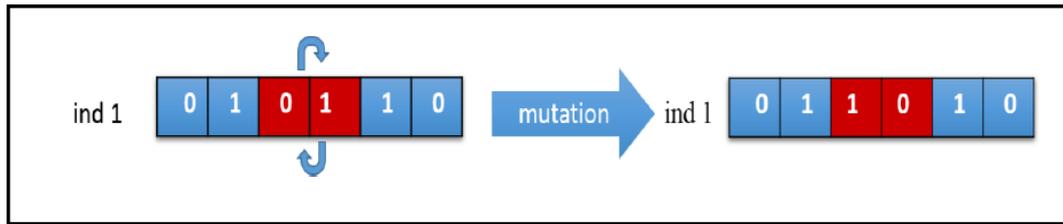


FIGURE 2.7: Une mutation binaire

Une version classique d'un algorithme génétique peut être décrite comme suit :

Algorithme 1.2 - Génétique [28]

1. Générer aléatoirement une population initiale P de N individus,
2. Evaluer chaque individu de la population P ,
3. Sélectionner des individus pour la reproduction,
4. Appliquer l'opérateur de croisement sur les individus sélectionnés,
5. Appliquer l'opérateur de mutation sur les enfants obtenus,
6. Evaluer l'adaptation des enfants obtenus et des mutants,
7. Sélectionner des parents qui seront remplacés par leurs enfants,
8. Effectuer le remplacement pour obtenir une population de N individus,
9. Répéter les étapes de 3. à 8. jusqu'à ce que le critère d'arrêt fixé soit atteint.

2.3 Conclusion

Dans ce chapitre, nous avons donné des rappels sur l'optimisation combinatoire et présenté le problème du voyageur de commerce qui est difficile à résoudre. La formulation mathématique du PVC, ainsi que les différentes méthodes de résolution ont été abordé dans ce chapitre. L'application de ce modèle dans les réseaux de capteurs sans fil sera décrit dans le chapitre suivant.

CHAPITRE 3

MODÉLISATION ET RÉOLUTION DU PROBLÈME

3.1 Introduction

Dans un réseau de capteurs, l'énergie est la ressource la plus précieuse qui influe directement sur la durée de vie des capteurs et du réseau entier [37]. Ceci est dû à la limitation de la capacité énergétique des batteries qui alimentent les capteurs d'une part et la forte possibilité de ne pas remplacer ces batteries d'autre part. Maximiser la durée de vie du réseau revient donc à réduire la consommation énergétique des noeuds. Malgré la diversité des protocoles à basse consommation d'énergie qui ont été proposés, l'énergie dans un réseau de capteurs reste toujours un problème de recherche ouvert, ce qui nécessite d'autres solutions qui viennent renforcer et améliorer les solutions existantes.

Dans notre travail, nous avons combiné le clustering et le principe du problème de voyageur de commerce (PVC) et nous avons appliqué la méthode de résolution approchée PPV pour préserver l'énergie des noeuds capteurs et prolonger la durée

de vie du réseau en conséquence. Dans ce chapitre, nous allons d'abord présenter le protocole LEACH, décrire aussi notre modèle mathématique et ensuite, nous détaillons notre proposition .

3.2 Consommation d'énergie dans les RCSFs

L'énergie consommée par un capteur est principalement due aux opérations suivantes : la détection, le traitement et la communication [40].

3.2.1 Energie de capture

Les sources de consommation d'énergie des noeuds par rapport aux opérations de détection ou de capture sont : l'échantillonnage, la conversion analogique-numérique, le traitement de signal et l'activation de la sonde de capture [41].

3.2.2 Energie de traitement

L'énergie de traitement est composée de deux sortes d'énergie : l'énergie de commutation et l'énergie de fuite. L'énergie de commutation est déterminée par la tension d'alimentation et la capacité totale commutée au niveau logiciel (en exécutant un logiciel). Par contre, l'énergie de fuite correspond à l'énergie consommée lorsque l'unité de calcul n'effectue aucun traitement. En général, l'énergie de traitement est faible par rapport à celle nécessaire pour la communication.

3.2.3 Energie de communication

L'énergie de communication se décline en deux parties : l'énergie de réception, l'énergie de l'émission . Cette énergie est déterminée par la quantité des données à communiquer et la distance de transmission, ainsi que par les propriétés physiques du module radio. L'émission d'un signal est caractérisée par sa puissance,

quand la puissance d'émission est élevée le signal aura une grande portée et l'énergie consommée sera plus élevée. Notons que l'énergie de communication représente la portion la plus grande de l'énergie consommée par un noeud capteur [33].

3.3 Techniques de minimisation de la consommation d'énergie

Nous présentons dans ce qui suit les différentes techniques utilisées pour minimiser la consommation. Ces techniques sont appliquées soit au niveau de la couche liaison soit au niveau de la couche réseau. Le schéma suivant donne un aperçu global de ces mécanismes :

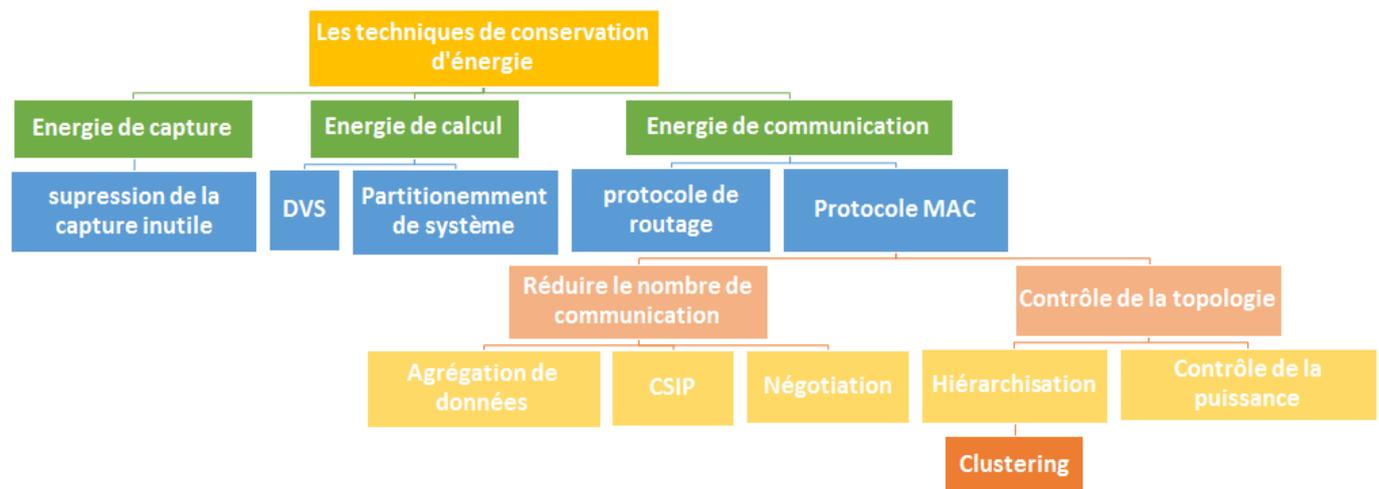


FIGURE 3.1: Les techniques de conservation d'énergie [30].

L'énergie du capteur peut être économisée soit au niveau de la capture, au niveau de traitement ou au niveau de la communication :

- A. La seule solution apportée pour la minimisation de la consommation d'énergie au niveau de la capture consiste à réduire les durées de captures.

- B. L'énergie de calcul peut être optimisée en utilisant deux techniques :
- L'approche DVS (Dynamique Voltage Scaling) [46] qui consiste à ajuster de manière adaptative la tension d'alimentation et la fréquence du microprocesseur pour économiser la puissance de calcul sans dégradation des performances.
 - L'approche de partitionnement de système qui consiste à transférer un calcul prohibitif en temps de calcul vers une station de base qui n'a pas de contraintes énergétiques et qui possède une grande capacité de calcul [34].
- C. La minimisation de la consommation d'énergie pendant la communication est étroitement liée aux protocoles développés pour la couche réseau et la sous-couche MAC. Ces protocoles se basent sur plusieurs techniques : l'agrégation de données, la négociation et la technique CSIP (Collaborative Signal and Information Processing). Cette dernière technique est une discipline qui combine plusieurs domaines [35] : la communication et le calcul à basse puissance, le traitement de signal, les algorithmes distribués, la tolérance aux fautes, les systèmes adaptatifs et la théorie de fusion des capteurs et des décisions.

Ces techniques ont le but de réduire le nombre d'émission et réception des messages.

3.4 Les principaux protocoles de routage dans les RCSF

3.4.1 Routage plat

Dans de nombreuses applications de réseaux de capteurs, vu le nombre élevé de noeuds déployés, il n'est pas possible d'attribuer des identificateurs globaux à chaque noeud. Cette absence d'identification globale avec le déploiement aléatoire

de noeuds capteurs font qu'il est difficile de sélectionner un ensemble spécifique de noeuds capteurs à interroger. Par conséquent, les données sont généralement transmises de chaque noeud capteurs dans la région de déploiement avec une redondance importante. Cette réflexion a conduit au routage data-centric [11] qui est différent du routage traditionnel où les routes sont créées entre les noeuds adressables gérées dans la couche réseau. Le destinataire envoie des requêtes à certaines régions et attend à recevoir des données provenant des capteurs situés dans les régions sélectionnées.

Comme les données sont demandées à travers des requêtes, le nommage est nécessaire pour préciser les propriétés des données [6].

3.4.2 Routage géographique

Les protocoles utilisant ce type de routage sont basés sur la localisation par GPS [6], et utilisent les informations d'emplacement des noeuds pour guider la découverte de routage ; ainsi que la transmission des données. Ils permettent la transmission directionnelle de l'information en évitant l'inondation d'information dans l'ensemble du réseau. Par conséquent, le coût de contrôle de l'algorithme est réduit et le routage est optimisé. De plus, avec la topologie réseau basée sur des informations de localisation de noeuds, la gestion du réseau devient simple. L'inconvénient de ces protocoles de routage est que chaque noeud doit utiliser le GPS pour savoir les emplacements des autres noeuds donc le réseau va consommer une très grande énergie.

3.4.3 Routage hiérarchique

L'objectif principal du routage hiérarchique [41, 8] est de maintenir efficacement la consommation d'énergie de noeuds capteurs en les impliquant dans la communication multi-sauts au sein d'un cluster et en effectuant l'agrégation et la

fusion des données afin de diminuer le nombre de messages transmis à la destination. La formation de clusters est généralement fondée sur la réserve d'énergie des capteurs et sur les capteurs qui sont à proximité du cluster-Head (voir figure 3.2). LEACH (Low Energy Adaptive Clustering Hierarchical) [9] est l'une de premières approches de routage pour les réseaux de capteurs. L'idée proposée par LEACH a été une inspiration pour de nombreux protocoles de routage hiérarchique, bien que certains protocoles ont été développés de manière indépendante.

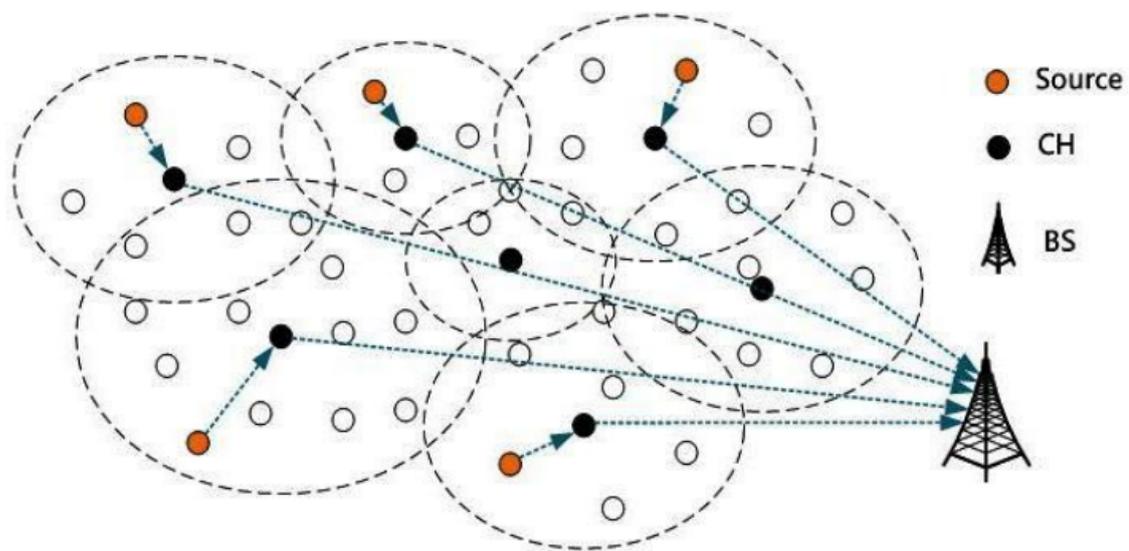


FIGURE 3.2: Topologie hiérarchique

3.5 Méthodes du groupement

Après un déploiement aléatoire des noeuds, le réseau suit un algorithme de clustering qui organise le réseau en sous réseaux (clusters), plus homogènes selon une métrique ou une combinaison de métriques, formant ainsi une topologie virtuelle. Chaque cluster identifie un noeud particulier appelé cluster-head. Ce dernier, permet de coordonner entre les noeuds membres de son cluster, d'agréger leurs données collectées et de les transmettre par la suite à la station de base.

De ce fait, seulement les noeuds CHs seront responsables de l'acheminement de l'information collectée vers la station de base.

Parmi les méthodes du groupement qui sont développées on cite :

3.5.1 Méthode de K-moyennes

Le partitionnement en k-moyennes (k-means) [32] est une méthode de partitionnement de données et un problème d'optimisation combinatoire. Étant donné des points et un entier K , le problème est de regrouper les points en K groupes homogènes, appelés clusters, de façon à minimiser une certaine métrique qui est souvent la distance. On considère la distance d'un point à la moyenne des points de son cluster. La fonction à minimiser est la somme des carrés de ces distances. Étant donné un ensemble de points (x_1, x_2, \dots, x_n) , on cherche à partitionner les n points en K ensembles $S = \{S_1, S_2, \dots, S_K\}$ ($K \leq n$) en minimisant la distance entre les points à l'intérieur de chaque partition :

$$\arg \min \sum_{i=1}^n \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

Où, μ_i est le barycentre des points dans S_i définit comme suit :

$$\mu_i = \frac{1}{|S_i|} \sum_{i \in S_i} x_i \quad (3.1)$$

3.5.2 Le choix du cluster-head

Le choix du cluster-head se fait dans la phase d'installation, selon une métrique (L'énergie résiduelle, la distance, ...) ou une combinaison de métriques.

3.5.3 Communication intra-cluster et inter-cluster

Chaque cluster-head est chargé d'assurer une bonne communication au sein de son cluster. Par ailleurs, il doit assurer le routage entre les différents cluster-

Algorithm 2: K-means.

- 1 On prend K nombre de centroïdes initialement à des endroits aléatoires.;
 - 2 Calculez la distance euclidienne de chaque nœud à tous centroïdes et attribuez-le au centroïde le plus proche.;
 - 3 Recalculez les positions des centroïdes dans chaque groupe et vérifiez le changement de position par rapport au précédent.;
 - 4 S'il y a un changement de position du centroïde, passez à l'étape 2, sinon les clusters sont finalisés et le processus de clustérisation se termine.
-

heads en maintenant les informations liées au routage. De plus, étant donné que les cluster-heads ne sont pas directement reliés, des noeuds passerelles doivent être choisis pour assurer la communication entre les cluster-heads.

3.5.4 La maintenance des clusters

Une mise à jour des clusters est réalisée soit dans le but de s'adapter aux changements de la topologie (dans le cas où un cluster-head ou un membre du cluster migre d'un cluster à un autre) soit pour équilibrer la consommation d'énergie dans le réseau (si le cluster-head garde son statut le plus longtemps possible, sa batterie sera épuisée, et ainsi il perdra son rôle). Parmi les protocoles proposés dans cette catégorie : LEACH [41].

3.6 Protocole LEACH

LEACH (Low Energy Adaptive Clustering Hierarchy) est un protocole de routage hiérarchique introduit par heinzelman et all [41]. Ce protocole se base sur le principe du "clustering" dans lequel les noeuds du réseau sont structurés suivant deux niveaux : les Cluster-Head (CHs) et les noeuds membres. LEACH est un protocole de routage réparti, proactif et dynamique. Les CHs sont choisis aléatoirement selon une politique de gestion round-robin afin de

garantir un équilibrage de la consommation énergétique entre les différents noeuds du réseau. Notons également, que dans LEACH, les CHs peuvent agréger les données reçues de plusieurs noeuds membres afin de réduire la quantité de données transmises à la station de base. Dans LEACH, chaque round est composé d'une phase d'installation et d'une phase de communication.

- **Phase d'installation** : Dans la phase d'installation, les CHs sont élus et une politique d'accès au médium est établie au sein de chaque groupe. Cette phase démarre par une prise de décision locale au niveau de chaque noeud afin de devenir CH ou membre. Pour ce faire, chaque noeud capteur n du réseau va choisir un nombre aléatoire, et si ce nombre est inférieur à une valeur donnée $T(n)$, alors le noeud en question devient CH. L'équation suivante permet de définir la valeur de $T(n)$ [41].

$$T(n) = \begin{cases} \frac{p}{1 - p * \left(r \bmod \frac{1}{p} \right)} & \text{Si } n \in G \\ 0 & \text{Sinon} \end{cases} \quad (3.2)$$

où

- p : représentant le pourcentage désiré de CH pendant un numéro de round r .
- G : représente l'ensemble des noeuds qui n'ont pas été élus CH pendant les $1/p$ rounds précédents.

Après cette phase d'installation, chaque noeud qui est élu CH va émettre un message de notification afin de permettre aux noeuds non CH de décider l'appartenance ou non à son cluster. Cette décision d'appartenance à un cluster est basée sur l'amplitude du signal reçu. En faisant ainsi, le CH

ayant le signal le plus fort est choisi, c'est-à-dire, le plus proche pour un noeud donné. En cas d'égalité, un chef aléatoire est choisi [33].

- **Phase de communication** : Durant la phase de communication, les noeuds membres peuvent émettre leurs données capturées en utilisant un ordonnancement avec des slots TDMA [41]. Cette technique de transmission permet ainsi aux noeuds membres d'éteindre leur radio en dehors de leurs slots réservés dans le but d'économiser leur énergie. Les communications entre les noeuds CHs et la station de base se font d'une manière directe. Ainsi, les CHs qui veulent émettre doivent adapter leurs émetteurs radio afin d'atteindre directement la station de base. Cependant, notons que le principal inconvénient de LEACH est que les CHs envoient directement les données vers la station de base, ce qui peut engendrer une forte consommation énergétique pour ces CHs.

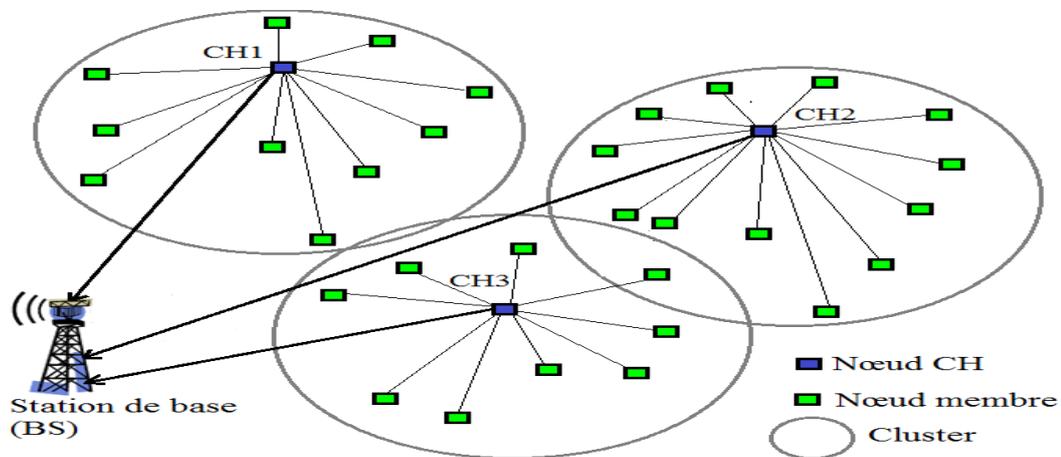


FIGURE 3.3: Architecture du routage hiérarchique LEACH.

3.7 Modèle de mobilité du puits proposé

3.7.1 Description du problème

Dans notre travail, nous avons traité le problème du routage dans les réseaux de capteurs sans fil, en minimisant la consommation d'énergie et améliorer la durée de vie du réseau. Pour cela, nous avons modifié notre problème comme étant un problème du voyageur de commerce, où un puits mobile représente le voyageur, et les CHs représentent les villes.

3.8 Le modèle mathématique

Notre objectif est de minimiser la distance parcourue du puits tout en passant une et une seule fois par chaque cluster head .Considérons la distance entre les clusters heads i et j est d_{ij} , tels que les coordonnées de deux clusters heads sont : $CH_i(x_i, y_i)$ et $CH_j(x_j, y_j)$. La matrice des distances entre les différents CHs est la suivante :

$$D = \begin{bmatrix} 0 & d_{12} & \dots & d_{1j} & \dots & d_{1K} \\ d_{21} & 0 & \dots & d_{2j} & \dots & d_{2K} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{i1} & d_{i2} & \dots & 0 & \dots & d_{iK} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{K1} & d_{K2} & \dots & d_{Kj} & \dots & 0 \end{bmatrix}$$

Avec,

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \quad i, j = \overline{1, K}. \quad (3.3)$$

1. **Les variables de décision :**

$$z_{ij} = \begin{cases} 1, & \text{Si le CH } j \text{ suit immédiatement le CH } i \text{ dans le parcours;} \\ 0, & \text{Sinon.} \end{cases}$$

2. Les contraintes :

(a) Le nombre de clusters heads que le puits mobile visite après le cluster head i est 1.

$$\sum_{j=1}^K z_{ij} = 1, \quad i = 1 \dots K. \quad (3.4)$$

(b) Le nombre de clusters heads que le puits mobile visite avant le cluster head j est 1.

$$\sum_{i=1}^K z_{ij} = 1, \quad j = 1 \dots K. \quad (3.5)$$

(c) La contrainte de sous-tournée : Soit S un sous ensemble de clusters heads visitées.

$$\sum_{i \in S} \sum_{j \in S} z_{ij} \leq |S| - 1, \quad \emptyset \neq S \subset \{1, 2, \dots, K\}. \quad (3.6)$$

3. La fonction objectif :

$$\min Z = \sum_{i=1}^K \sum_{j=1}^K d_{ij} z_{ij} \quad (3.7)$$

4. Le modèle mathématique :

$$\left\{ \begin{array}{l}
 \min Z = \sum_{i=1}^K \sum_{j=1}^K d_{ij} z_{ij} \\
 \sum_{j=1}^K z_{ij} = 1, \quad i = 1 \dots K \\
 \sum_{i=1}^K z_{ij} = 1, \quad j = 1 \dots K \\
 \sum_{i \in S} \sum_{j \in S} z_{ij} \leq |S| - 1, \quad \emptyset \neq S \subset \{1, 2, \dots, K\} \\
 z_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, K
 \end{array} \right. \quad (3.8)$$

3.9 Solution proposée

Notre but est la conservation de l'énergie des noeuds capteurs pour prolonger la durée de vie du réseau. Pour cela, nous avons conçu une solution qui se base sur le principe du problème de voyageur de commerce en utilisant la méthode de résolution approchée PPV. Dans un premier temps, nous allons présenter les hypothèses sur laquelle notre proposition se base.

3.9.1 Hypothèses

Pour appliquer notre approche, nous nous basons sur le modèle du réseau suivant :

- Le réseau de capteur est statique (Les noeuds ne sont pas mobiles).
- Les noeuds capteurs sont tous identiques (même énergie initiale, même capacité de stockage et de traitement de données).
- Les noeuds sont distribués sur la zone de capture aléatoirement.
- Le réseau est organisé en cluster à l'aide d'un algorithme de clustering où, chaque cluster a un cluster head (CH).
- La surface de simulation est en 2D (deux dimensions X et Y) et la distance en mètres.
- Les liens de communication entre les noeuds sont des liens bidirectionnels.
- Le puits mobile est un noeud qui a une trajectoire déterminé par le PVC.
- On considère que la station de base est fixée à une voiture, et son énergie est considérée illimitée.
- La distance fixée entre la station mobile et chaque CH est supposée petite.

3.9.2 Détail de la solution proposée

L'objectif de la solution proposée est la minimisation de la consommation d'énergie dans un réseau de capteurs sans fils, afin de prolonger la vie des noeuds et rendre le réseau opérationnel le plus long temps possible. Pour mettre en place notre solution, nous nous sommes basées sur la notion de voyageur de commerce utilisée dans l'optimisation combinatoire.

L'idée consiste à modéliser le réseau par un graphe qui contient uniquement les noeuds clusters chefs (CHs) et la station de base. Ensuite, on applique l'algorithme de PPV pour trouver la route du puits mobile. Nous rappelons que les clusters chefs sont les sommets de notre graphe non orienté.

La figure 3.4 illustre le schéma de la solution proposée.

Le fonctionnement de notre solution proposée est établi en trois phases : la phase d'installation (configuration) où, tous les clusters sont formés. Puis, le

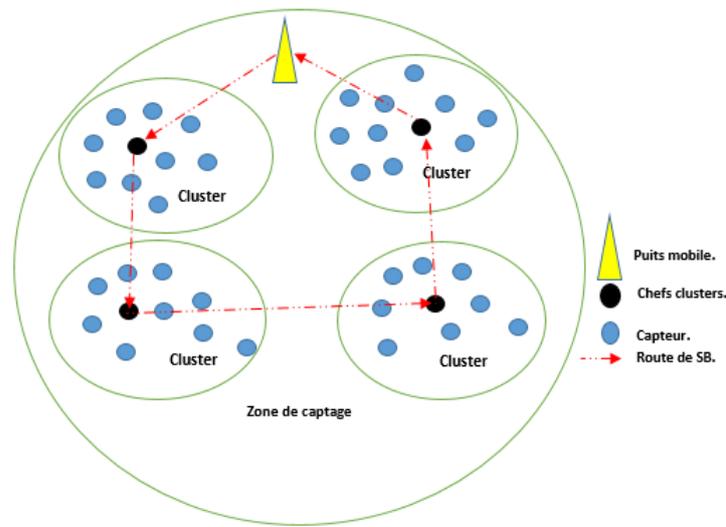


FIGURE 3.4: Schéma de la solution proposée.

modèle de mobilité. Enfin, la phase de communication où, la station de base communique avec les CHs pour collecter les données. Les différentes phases de l'algorithme proposé sont détaillées dans ce qui suit :

Phase d'installation :

Une fois que le RCSFs est crée on exécute la phase d'installation, qui s'exécute une seule fois durant tout le processus de routage. Afin de réaliser un regroupement (clustering) des noeuds de capteurs à l'aide de l'algorithme de K-means qui est décrit comme suit :

Algorithm 3: Phase d'installation.

Input : N, K

Output: CHs

- 1 Donner les caractéristiques des noeuds capteurs .
 - 2 Grouper les noeuds en K clusters en utilisant l'algorithme de K-means.
 - 3 Sélectionner les CHs, dans la première tournée : la métrique choisie est la distance entre les CHs.
-

Phase de routage

Après avoir sélectionné les clusters-heads, le puits mobile doit visiter tous les chefs du clusters une et une seule fois, ceci est équivalent à la recherche d'une tournée minimale du puits mobile (voyageur) en démarrant de sa position initiale de coordonnées (sbx, sby) , puis passe par chaque CH une et une seule fois et revenir à sa position initiale.

Soit $G = (V, A)$ un graphe composé d'un ensemble V de n Cluster-Heads plus le puits, et d'un ensemble A d'arêtes. Soit d_{ij} une matrice de distances relative à A . On cherche à trouver un circuit Hamiltonien, c'est-à-dire un circuit qui ne passe qu'une seule fois par chaque cluster-heads, de longueur minimale. Comme le problème est symétrique, donc chaque distance i vers j est la même que la distance de j vers i . Donc pour optimiser la route du puits on va utiliser la méthode de résolution approché *PPV* qui est décrit comme suit.

Algorithm 4: Modèle de mobilité.

Input : CLheads, SB

Output: longueurpluscourtchemin, pluscourtchemin

1 $S =$ l'ensemble des CHs ;

2 Noeud courant $c = SB$;

3 **while** $\exists CH \in S$ non visité **do**

4 //Sélectionner CH le voision le plus proche de la station de base
 tq $\forall CH, CH' \in S, d(c, CH) \leq d(c, CH')$;

5 $c = CH$;

6 **end**

7 **Aller de c vers SB ;**

Phase de communication

Dans cette phase, les noeuds capteurs peuvent commencer la détection et la transmission de données à leur chef, et ce dernier agrège ces données et attend l'arrivée de la station de base pour la collection des données comme il est décrit ci-dessous.

Algorithm 5: Phase de communication.

Input : $E_0, E_{elec}, E_{Tx}, E_{Rx}, E_{amp}, E_{DA}, S$.

Output: Energie moyenne, Energie consommée dans le réseau.

- 1 Dans la Phase de communication : la métrique choisie pour sélectionner les CHs est le noeud qui a la plus grande énergie.
 - 2 On calcule dans chaque cluster l'énergie de transmission pour les noeuds membres et l'énergie de réception pour les CHs.
 - 3 Après l'arrivée de la station de base aux CHs on calcule leurs énergie de transmission avec une distances fixe très petite.
-

3.10 Conclusion

Dans ce chapitre nous avons modélisé le problème posé et nous avons aussi détaillé notre solution proposée pour optimiser le chemin de routage à basse consommation d'énergie, pour les réseaux de capteurs sans fil (RCSFs).

Dans le chapitre suivant, on va simuler notre approche sur MATLAB.

CHAPITRE 4

EXPÉRIMENTATION NUMÉRIQUE

4.1 Introduction

Ce chapitre est consacré à la simulation du protocole LEACH et celui de notre approche, basé sur le PVC. Pour ce faire, on a implémenté notre approche sous l’environnement de simulation MATLAB. Pour comparer entre les deux protocoles, nous avons généré aléatoirement un RCSF de 1000 noeuds capteurs. Les métriques de comparaison utilisés sont la durée de vie du réseau ainsi que l’énergie consommé par tournée lors de la simulation.

4.2 Choix du simulateur

Il existe plusieurs simulateurs, parmi lesquels on retrouve :

- **NS-3 (Network Simulator)** : NS-3 [2] est un simulateur réseau, open source à événements discrets. Son noyau et ses modèles sont implémentés en C++, mais avec une interface de script Python. L’objectif de NS-3 est de développer un environnement de simulation ouvert pratique pour la re-

cherche dans les réseaux. Les outils `nam`, `ns3-viz` associés au simulateur NS permettent la visualisation des animations issues de la simulation, tels que le transfert et la taille des paquets échangés, l'état des files d'attente, etc ...

- **OMNeT++** : tout comme NS-3, OMNeT++ [3] est un simulateur réseau à événements discrets programmé en C++. Il peut être utilisé dans les deux types de réseaux filaires et sans fil.
- **MATLAB** : Développé par la société The MathWorks, MATLAB [1] est optimisé pour résoudre les problèmes scientifiques et techniques. C'est un langage basé sur les matrices qui combine les mathématiques, le graphisme et la programmation.

Notre choix c'est porté sur MATLAB, car son environnement est simple à utiliser et dispose d'une panoplie d'outils très puissante. Dans notre mémoire, on a utilisé la version R2012a de MATLAB.

4.3 Métriques de simulation

Dans les réseaux de capteurs sans fil, la plus part des approches visent à minimiser la consommation d'énergie et prolonger la durée de vie du réseau. Les métriques considérées dans ce travail sont les suivantes :

- **Consommation d'énergie** : représente la quantité d'énergie consommée par l'ensemble des noeuds dans le réseau durant le nombre de tournée du puits.
- **Durée de vie du réseau** : représente le nombre de noeuds vivants dans chaque tournée.

4.4 Paramètres de simulation

Le réseau généré est composé de $N = 1000$ noeuds, déployés aléatoirement sur une zone de capture d'une surface de dimension $80 \times 80 \text{ m}^2$. Dans notre cas, un réseau est codé par un tableau d'enregistrements de taille N , où chaque enregistrement représente un noeud capteur et le tableau suivant résume ces paramètres essentiels :

Description du champ	Paramètres
Numéro du noeud capteur	Id
Niveau d'énergie des noeuds	E_0
Une variable d'état indiquant si le noeud est vivant ($Cond = 1$), ou mort	<i>Cond</i>
Role = 1 si le noeud est un CH	<i>Role</i>
Coordonnées du noeud	(x, y)

TABLE 4.1: Caractéristiques d'un noeud.

Notons que les coordonnées d'un capteur sur un plan 2D dont deux valeurs aléatoires (x et y).

Les principaux paramètres de simulation d'un RCSFs sont résumés dans le tableau ci-après :

Description	Paramètres
Energie consommée en émission	E_{Tx}
Energie consommée en réception	E_{Rx}
Energie consommée par le transmetteur électronique	E_{elec}
Energie consommée par l'amplificateur de l'émetteur	E_{amp}
Energie nécessaire à l'agrégation des données	E_{DA}
La taille en bits d'un paquet de données	s
La distance entre deux noeuds	d

TABLE 4.2: Paramètre de simulation.

4.5 Etapes de simulation

Les étapes de simulation sont les suivantes :

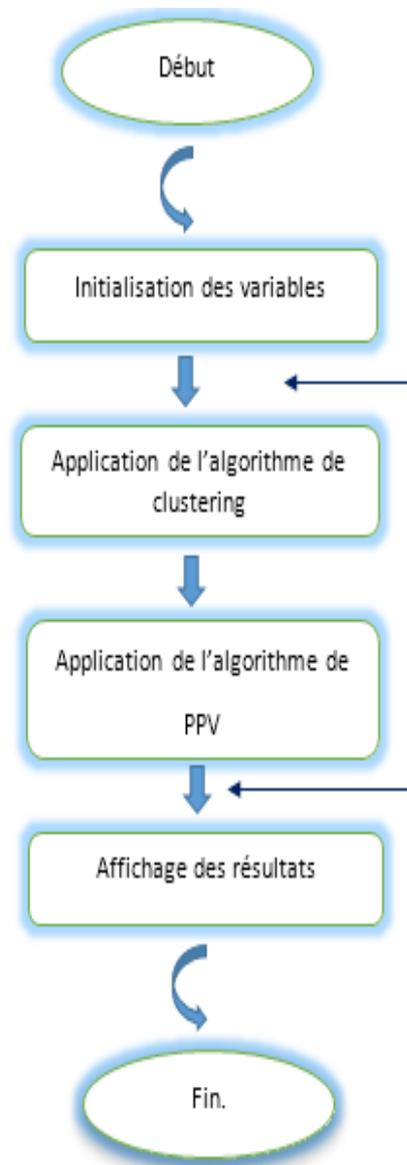


FIGURE 4.1: Etapes de simulation.

4.5.1 Initialisation des variables

Cette étape correspond à la déclaration des variables globales (nombre de capteurs, surface du terrain simulé, nombre de tests et d'exécution à réaliser. . .), leur initialisation, la création des capteurs (portée de transmission, capacité énergétique, capacité mémorielle, capacité calculatoire des capteurs) et leur déploiement aléatoire sur le terrain simulé. En effet, le déploiement des noeuds se fait d'une manière aléatoire sur une surface de capture. Chaque capteur dans le réseau est représenté par ses coordonnées (x, y) . On suppose qu'au début de

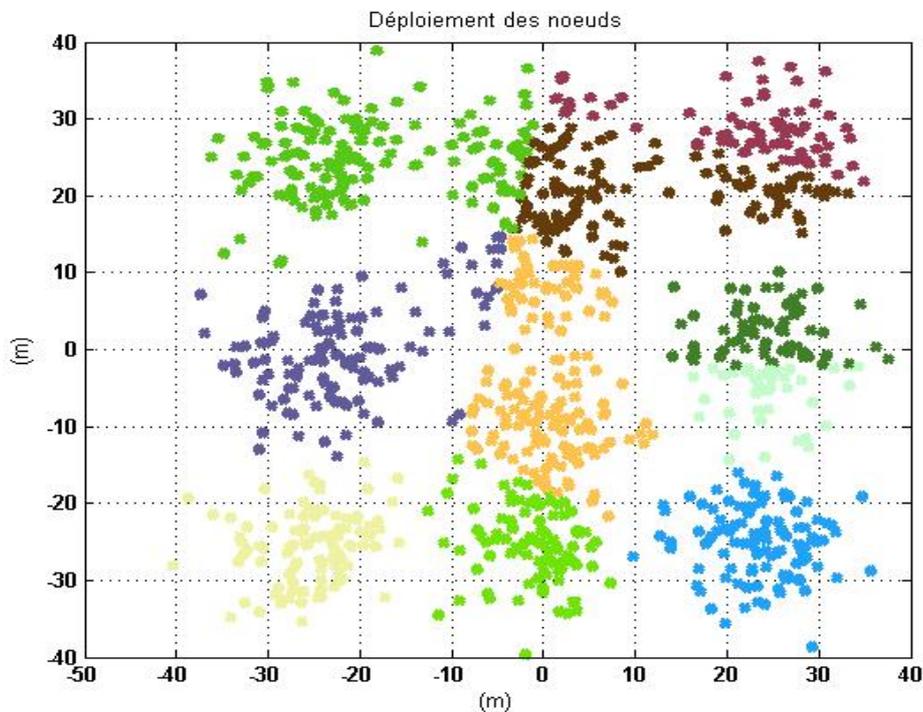


FIGURE 4.2: Déploiement des noeuds.

la simulation que tous les noeuds ont le même niveau d'énergie.

4.5.2 Application de l'algorithme de clustering

Cette étape concerne la détermination des groupes (clusters) et de sélectionner le chef (CH) de chaque cluster .

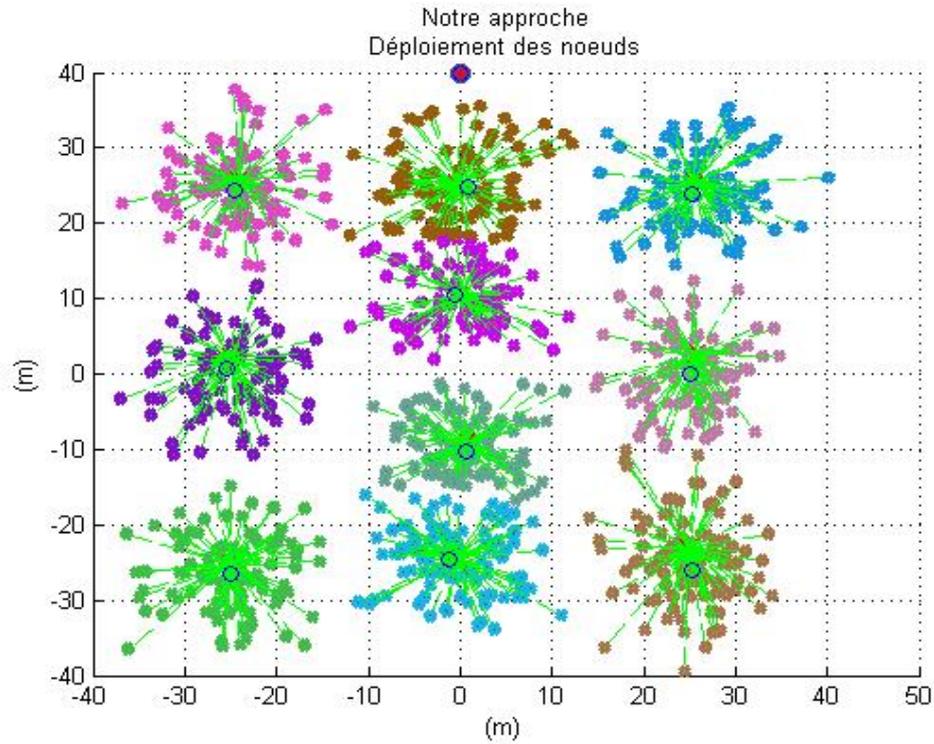


FIGURE 4.3: Groupement des noeuds .

4.5.3 Optimisation du chemin de routage

Trouver un chemin optimal pour le puits mobile est identique au problème du voyageur de commerce TSP . Tel que : un puits mobile représente le voyageur, et les chefs des clusters définissent les villes.

L'optimisation de la route du puits mobile pour visiter tous les clusters heads une et une seule fois est équivalente à la recherche du plus court voyage d'un commerçant pour visiter chaque ville une seule fois.

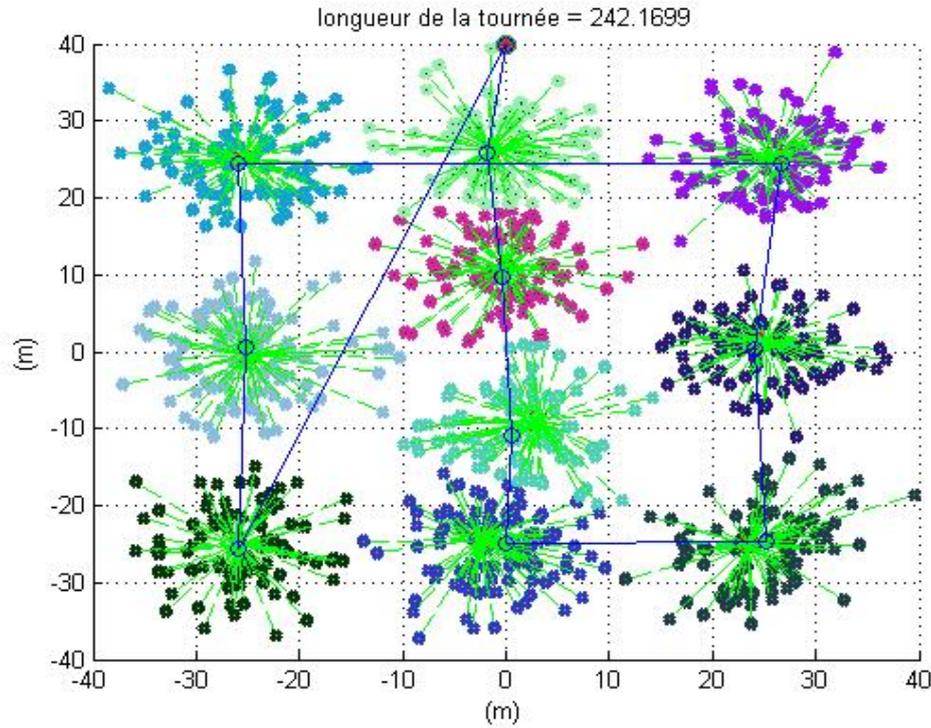


FIGURE 4.4: La tournée de la station de base.

4.5.4 Affichage des résultats

les courbes obtenues par rapport à notre approche implémentés, selon les critères d'évaluation de performances choisis (*Moyenne d'énergie consommée*, *Durée de vie du réseau*) seront présentés.

4.6 Modèle de consommation d'énergie

Le modèle radio (voir figure 4.3) proposé par Heinzelman et al [41] a été utilisé pour calculer l'énergie consommée en émission et en réception des messages.

Selon ce modèle l'énergie consommée pendant la transmission (E_{Tx}) d'un message de s bits est donnée par :

$$E_{Tx} = E_{elec} * s + E_{amp} * s * d^2 \quad (4.1)$$

Alors que l'énergie consommée pendant la réception (E_{Rx}) d'un message de s bits est donnée par :

$$E_{Rx} = E_{elec} * s \quad (4.2)$$

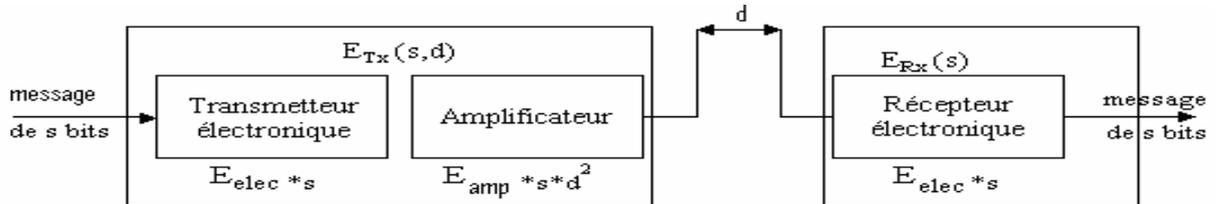


FIGURE 4.5: Modèle d'énergie utilisé [41].

4.7 Validation numérique

Dans cette expérience on a exécuté notre algorithme dans une machine équipée de : *un environnement Windows 8 professionnel, un processeur Intel(R) Core(TM) i5 CPU M540 et 4 GO de RAM*, les valeurs de paramètres de simulation qu'on a utilisés sont :

- Les coordonnées de la station de base sont :
 - $sb.id=11;$
 - $sb.x = 0 \text{ m};$
 - $sb.y = 40 \text{ m};$
- $N = 1000 \text{ noeuds};$
- $E_o = 0.01 \text{ Joules};$
- $E_{elec} = 50 * 10^{-9} \text{ Joules/bit};$
- $E_{Tx} = 50 * 10^{-9} \text{ Joules/bit};$
- $E_{Rx} = 50 * 10^{-9} \text{ Joules/bit};$
- $E_{amp} = 100 * 10^{-12}; \text{ Joules/bit/m}^2$

- $E_{DA} = 5 * 10^{-9}$; *Joules/bit*
- $s=1000$ bits;
- $K=10$ clusters;

4.7.1 Les résultats obtenus

Pour la configuration ci-dessus nous avons obtenue les résultats suivants :

Résultats	Notre approche	LEACH
La tournée où le premier noeud est mort	90 tournée	27 tournée
La tournée où tous les nœuds meurent	200 tournée	180 tournée
L'énergie moyenne des noeuds avant la mort du premier noeud	0,0056 J	0,1175 J
L'énergie moyenne des noeuds lorsque tous les nœuds meurent	0.0501 J	0,0558 J

TABLE 4.3: Tableau comparatif entre LEACH et notre approche.

4.8 Evaluation de performances

Dans ce qui suit, nous allons présenter, analyser, et comparer les résultats de simulations obtenus avec MATLAB de l'application de protocole LEACH et notre approche suivant les métriques discutées précédemment

4.8.1 Consommation d'énergie par tournée

La figure 4.6 illustre l'évolution de la consommation d'énergie des noeuds en fonction des tournées. Elle montre clairement qu'il y a une différence énergétique entre le protocole LEACH et notre approche. On remarque que l'énergie consommée durant chaque tournée du puits du réseau avec le protocole LEACH est plus élevée comparée à celle du notre approche.

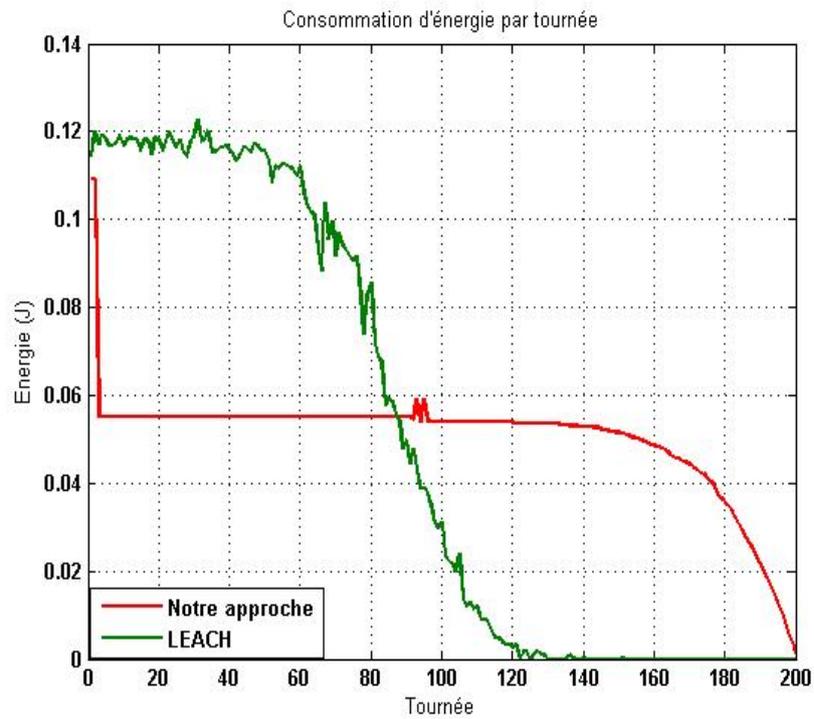


FIGURE 4.6: Consommation d'énergie par tournée.

4.8.2 Durée de vie du réseau

Pour pouvoir étudier la durée de vie du réseau en fonction du nombre de tournée, et d'examiner l'efficacité de notre protocole par rapport à celui de LEACH, nous avons suivi l'évolution de 1000 noeuds capteurs, comme représenté par la figure 4.7 .

Il est clair, que la durée de vie du réseau de notre protocole est mieux que celle du protocole LEACH. On remarque aussi que dans la 90 ème tournée le premier noeud épuise son énergie, par contre dans LEACH le premier noeud meurt dans la 27 ème tournée .

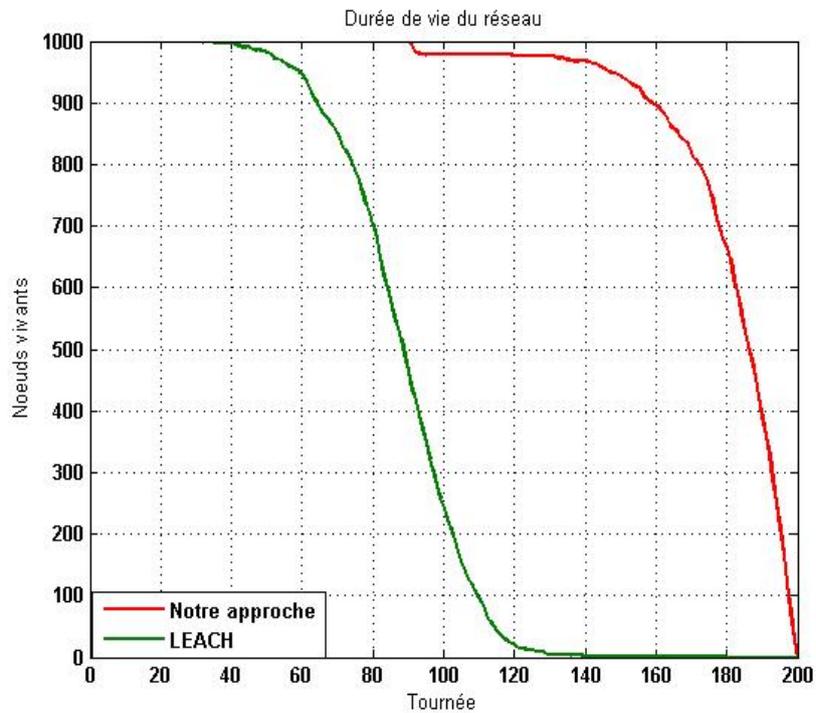


FIGURE 4.7: Durée de vie du réseaux en fonction de tournée.

4.9 Conclusion

Dans ce chapitre, nous avons présenté la technique de simulation de notre approche d'optimisation du chemin de routage sur MATLAB ainsi que les résultats obtenus après implémentation, il a été évalué en fonction d'énergie consommée, et la durée de vie du réseau.

CONCLUSION GÉNÉRALE

Les réseaux de capteurs sans fil ont connu un grand succès dès leur apparition dans tous les domaines (militaire, médicale, environnementale,..., etc.) grâce à leurs diverses fonctionnalités. Cependant, la conception de ces réseaux doit répondre aux contraintes de limitation de la ressource énergétique.

Dans notre travail, nous avons traité le problème du routage dans les réseaux de capteurs sans fil, en minimisant la consommation d'énergie et améliorer la durée de vie du réseau. Pour cela, nous avons modifié notre problème comme étant un problème du voyageur de commerce et nous avons utilisé l'heuristique de PPV pour la résolution.

D'après les résultats qu'on a obtenu dans la simulation, notre objectif est bien atteint qui est le prolongement de la durée de vie d'un RCSFs.

En guise de perspectives de ce travail, nous suggérons les points suivants :

- Utilisation de plusieurs puits mobiles.
- Etudier le cas du multi-critères de ce problème (par exemple, maximiser l'énergie résiduelle et minimiser la tournée du puits).

BIBLIOGRAPHIE

- [1] Matlab. <http://fr.mathworks.com/products/matlab>. Consulter le :21-05-2019.
- [2] Ns-3. <https://www.nsnam.org/overview/what-is-ns-3>. Consulter le :21-05-2019.
- [3] Omnet++. <https://omnetpp.org>. Consulter le :21-05-2019.
- [4] J. Rezazadeh A. S. Ismail and M. Moradi.
- [5] D.Patroumpa J.Rolim A.Kinalis and S.Nikoletseas. Biased sink mobility with adaptive stop times for low latency data collection in sensor networks. *Information Fusion*, 15 :56–63, 2012.
- [6] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Journal of Ad Hoc Networks*, 3(3) :325–349, 2005.
- [7] M. Akli. Problème de tournée et de véhicules avec contraintes et fentre de temps. Thèse de magister, Université Mouloud Mammeri de Tizi Ouzou, 2013.
- [8] J.N. Al-Karaki and A.E. Kamal. A survey routing techniques in wireless sensor networks :. *IEEE Wireless Communications*, 2004.

- [9] M. Ali and S. K. Ravula. Real-time support and energy efficiency in wireless sensor networks. *Technical report*, January 2008.
- [10] K. Amari. Elaboration d'un algorithme génétique hybride pour l'optimisation des puissances actives dans un réseau d'énergie électrique. Mémoire d'ingénieur, Université Mohamed Boudiaf d'Oran, 2012.
- [11] D. Estrin B. Krishnamachari and S. Wicker. Modeling data-centric routing in wireless sensor networks. *USC Computer Engineering, Technical Report CENG*, pages 02–14, 2002.
- [12] N. Barnier and P. Brisset. Optimisation par algorithme génétique sous contraintes. *Article*, 18, 1999.
- [13] I. Boussaid. Perfectionnement de métaheuristiques pour l'optimisation continue. Thèse en cotutelle de doctorat en science, Université Paris-Est Créteil et USTHB, 2013.
- [14] C. Srisathapornphat C. C. Shen and C. Jaikaeo. Sensor information networking architecture and applications. *IEEE Personal Communications*, 8 :52 – 59, 2001.
- [15] M.Srivastava C.Schurgers, S.Ganeriwal and V.Tsiatsis. Optimizing sensor networks in the energy-latency-density design space. *IEEE Trans. Mob. Comput*, 1 :70 – 80, 2002.
- [16] E.H.SMarour Diop. *Optimisation de la transmission d'image dans les réseaux de capteurs*. Thèse en cotutelle de doctorat en science, université de Pau France et université Gaston Berger de saintlouis Sénégal, juin 2014.
- [17] H. Lakhbab S. Douiri and S. Elbernoussi. Cours des méthodes de résolution exactes heuristiques et métaheuristiques.
- [18] H. Land et A. G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28 :497 – 520, 1960.

- [19] M. Khelaf et A. Zaidi. Application d'un algorithme génétique pour générer le prix d'un actif boursier. Master's thesis, Université A.MIRA de Béjaia, 2007.
- [20] S. K. Singh et al. Applications, classifications and selections of energy-efficient routing protocols for wireless sensor networks. *International Journal of advanced engineering sciences and technologies (IJAEST)*, 1 :85–95, 2011.
- [21] D.Allouache et K.Azamoum. Optimisation de la consommation d'énergie dans les réseaux de capteurs sans fil. Mémoire de master, Université A.MIRA-BEJAIA, 2013-2014.
- [22] M.Yildizoglu et T. Vallée. Présentation des algorithmes génétiques et de leurs applications en économie. *Article universitaire*, 2001.
- [23] W.Zhao E.Zegura and M.Ammar. A message ferrying approach for data delivery in sparse mobile ad hoc networks. *In Proceedings of the 5th ACM international symposium on Mobile Ad Hoc networking and computing*, pages 187 – 198, may 2004.
- [24] P. Fouilhoux. Optimisation combinatoire : Programmation linéaire et algorithmes. Master's thesis, Université Pierre et Marie Curie, 2015.
- [25] M.Di Francesco G.Anastasi and S.K.Das. Data collection in wireless sensor networks with mobile elements : A survey. *ACM Transactions on Sensor Networks*, 8, 2011.
- [26] V. Gardeux. *Conception d'heuristiques d'optimisation pour les problèmes de grande dimension. Application à l'analyse de données de puces à ADN*. Thèse de doctorat en informatique, Université de Paris-Est Créteil, 2011.
- [27] M.I.Khan G.Haring and W.N.Gansterer. Congestion avoidance and energy efficient routing protocol for wireless sensor networks with a mobile sink. *J. Netw*, 2 :42 – 49, 2007.

- [28] A. Gherboudj. *Méthodes de résolution de problèmes difficiles académiques*. Thèse de doctorat en informatique, Université de Constantine 2, 2013.
- [29] M. Ilyas and I. Mahgoub. *Handbook of Sensor Networks Compact Wireless and Wired Sensing Systems*. Number 08493196864 in CRC PRESS LLS. USA, 2005.
- [30] S. MOAD. La consommation d'énergie dans les réseaux de capteurs sans fil. 2008.
- [31] J.P.Hubaux J.Luo M.Piorkowski, M.Grossglauser and J.Panchard. Mobicroute : Routing towards a mobile sink for improving life time in sensor networks. *In Proceedings of the Distributed Computing in Sensor Systems*, pages 480 – 497, 2006.
- [32] A. Likas N. Vlassis and J. J.Verbeek. the global k-means algorithm. *Pttern recognition*, 36(2) :pp. 451–461, 2003.
- [33] D. NGOM. *Optimisation de la durée de vie dans les réseaux de capteurs sans fil sous contraintes de couverture et de connectivité*. PhD thesis, Université de Haute Alsace (France) et l'Université Cheikh Anta Diop de Dakar (Sénégal), 2016.
- [34] PaoloSanti. Topology control in wireless ad hoc and sensor networks. *Hardcover*, 2005.
- [35] D. Shepherd S. Kumar and F. Zhao. Collaborative signal and information processing in micro-sensor networks. *IEEE Signal Processing Magazine*, 2002.
- [36] I.Chatzigiannakis S. Nikolettseas and A. Kinalis. Efficient data propagation strategies in wireless sensor networks using a single mobile sink. *Comput. Commun*, 31 :896 – 914, 2008.

- [37] S. Rath B-P-S. Sahoo and D. Puthal. Energy efficient protocols for wireless sensor networks : A survey and approach. *International Journal of Computer Applications*, 44 :43–48, 2012.
- [38] I.Chatzigiannakis S.Nikoletseas and A.Kinalis. Sink mobility protocols for data collection in wireless sensor networks. *In Proceedings of the 4th ACM International Workshop on Mobility Management and Wireless Access*, pages 52–59, october 2006.
- [39] R. TOUATI. *Reconnaissance des actions humaines à partir d’une séquence vidéo*. Thèse de doctorat en informatique, Université de Montréal, 2014.
- [40] S. Park V. Raghunathan, C. Schurgers and M. B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2) :40–50, 2002.
- [41] A. Chandrakasan W. R. Heinzelman and H. Balakrishnan. Energy-efficient communication protocol for wireless micro-sensor networks. *In Proceedings of the 33 rd annual Hawaii International Conference*, page 10, 2000.
- [42] Wikipedia. Exemple de permutation 2-opt. <https://fr.wikipedia.org/wiki/2-opt>. Consulter le :21-05-2019.
- [43] W. Su Y. Sankarasubramaniam, I.F. Akyildiz and E. Cayirci. : A survey on wireless sensor networks. 38(4) :393–422, 2002.
- [44] Y.CHALLAL. Sécurité des réseaux de capteurs sans fil. *Techniques de l’ingénieur Cryptographie, authentification, protocoles de sécurité, VPN*, base documentaire : TIB314DUO(ref. article : h5390), 2015.
- [45] Y. Younes. Minimisation d’énergie dans un réseau de capteur. mémoire de magister, Université Mouloud Mammeri de Tizi Ouzou, septembre 2012.
- [46] S. Ziane and A. Mellouk. A swarm intelligent scheme for routing in mobile ad hoc networks. *IEEE Systems Communications*, 2005.

Résumé

La consommation d'énergie dans les RCSFs est l'un des problèmes étudiés actuellement dans la littérature. De ce fait, la mobilité de la station de base est l'une des solutions proposées pour prolonger leurs durées de vie.

L'objectif de notre travail est d'appliquer les techniques de la recherche opérationnelle dans le domaine des RCSFs avec une station de base mobile. Afin de prolonger la durée de vie du réseau, la stratégie adoptée par le puits est celle d'un voyageur de commerce.

Mots-clés : RCSF, mobilité du puits, PVC, méthode du PPV, clustering par K-moyennes, protocole LEACH, simulation avec MATLAB.

Abstract

The energy consumed in the WSNs is one of the problems studied in the literature. As a result, the mobility of the base station is one of the solutions proposed to extend their life-time.

The goal of our work is to master operational research techniques in the field of WSNs with a mobile base station. In order to extend the life of the network, the well adoption strategy is that of a commercial traveler.

Keywords: WSN, mobility of the sink, TSP, method of NN, clustering with K-means, protocol LEACH, simulation with MATLAB.