



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Réseaux et Télécommunications

Par :

BEKKI Mimouna
BOUAIT Nadia

Sur le thème

Un algorithme de colonies de fourmis pour la couverture de cibles dans les réseaux de capteurs

Soutenu publiquement le 31/ 10 /2019 à Tiaret devant le jury composé de :

Mr NASSANE Samir	M.A.A Université IBN-KHALDOUN Tiaret	Président
Mr BENGHENI Abdelmalek	M.A.A Université IBN-KHALDOUN Tiaret	Encadreur
Mr BEKKAR Khaled	M.A.A Université IBN-KHALDOUN Tiaret	Examinateur

Année Universitaire : 2018 -2019

Remerciements

Nous tenons en premier lieu à remercier Allah qui nous a éclairé le chemin du savoir et qui nous a donné la volonté et la patience d'achever ce modeste travail.

Nous adressons nous plus vifs remerciements

À notre promoteur MR. BENGHANI Abdelmalek qui nous a fait l'honneur d'avoir accepté la charge de nous diriger notre travail avec une grande patience, pour la confiance qu'il a eu en notre projet et surtout pour ses orientations.

À messieurs les membres de jury de ce mémoire: Mr. BAKKAR Khaled et Mr. NAASSANE Samir pour avoir bien voulu examiner ce travail.

À tous les enseignants qui ont participé à la formation tout au long notre cycle Universitaire jusqu'aux jours d'aujourd'hui.

Enfin, nous ne voudrions pas non plus oublier toutes les personnes que nous pouvons rencontrer au long de ces années universitaire tous ceux qui ont contribué de près ou de loin à l'accomplissement de ce travail.

Dédicaces

*Je remercie Allah de m'avoir donné le courage pour accomplir
ce modeste travail que je dédie :*

À mes très chers parents qui sont la bougie qui illumine ma vie

À mon frère Omar et sa femme

À mes sœurs Fatima et Meriem

À ma grande mère

À mes très chères amies

À ma grande famille, à tous mes Enseignants

Et à tous ceux que j'aime et qui m'aiment.

BOUAIT NADIA

Dédicaces

Je dédie ce modeste travail tout d'abord :

À celui qui m'a aidé et m'a encouragé durant mes études sur tous mes chers parents qui ont sacrifié tout pour que je puisse faire ce petit effort et qui m'ont aidé avec leur suivis, leur soutiens et leur souhaits de réussite durant toute ma vie.

Je dédie aussi ce travail :

*À mes frères Ahmed et sa femme, Miloud, Rabah et sa femme,
À mes sœurs Fatima, Nadjet et Sara pour leurs encouragements
qui seront fières de ce travail,
À ma grande famille pour le grand support,
À mes chères amies et toutes les personnes que j'aime et en qui je
les ai rencontrés durant toute ma vie sans oublier bien sur toute
ma promotion.*

BEKKI MIMOUNA

Résumé :

La couverture dans les réseaux de capteurs sans fil (RCSF) est souvent considérée comme étant une mesure de performance très importante. Elle reflète la façon dont une zone donnée est surveillée (contrôlée), c'est-à-dire comment chaque point de la zone de surveillance est observé et suivi par l'ensemble des nœuds. Cette zone est souvent présentée comme un cercle de rayon r d'où le nœud à une vision qui dépend du rayon de réception de son module de communication r_s . La couverture d'un RCSF mesure le degré de couverture de l'espace d'intérêt par l'ensemble des nœuds capteurs. Cette couverture peut exister en trois sortes différentes: la couverture partielle, la couverture dense et la couverture redondante. Par ailleurs, un mauvais déploiement des nœuds capteurs produira une perte de certains nœuds qui n'auront pas de voisins et qui seront isolés et déconnectés du réseau. Ainsi, une meilleure couverture est la clé pour les systèmes robustes et peut-être exploitée pour étendre significativement la durée de vie du RCSF en basculant les nœuds redondants en mode veille de faible consommation d'énergie. L'objectif de notre travail dans ce mémoire est d'utiliser un algorithme de colonies de fourmis appelé API-Apicals (Pachycondyla-apicalis) pour optimiser la couverture de cibles dans le RCSF tous en assurant les différentes contraintes du RCSF telles que: la connectivité dans le RCSF et la durée de vie du RCSF.

Mots clés : Réseaux de capteurs sans fil, Couverture, connectivité, l'algorithme de colonie de fourmi.

ملخص:

غالبًا ما تعتبر التغطية في شبكات الاستشعار اللاسلكية مقياسًا مهمًا جدًا للأداء. إنها تعكس الطريقة التي يتم بها مراقبة منطقة معينة (التحكم فيها)، أي كيفية ملاحظة كل نقطة في منطقة المراقبة وتتبعها جميع العقد. يتم تقديم هذه المنطقة غالبًا على أنها دائرة نصف قطرها r حيث يكون للعقدة رؤية تعتمد على نصف قطر الاستقبال لوحدة الاتصالات r_s . تقيس تغطية شبكة درجة تغطية الفضاء ذي الأهمية من خلال جميع عقد الاستشعار. يمكن أن توجد هذه التغطية في ثلاثة أنواع مختلفة: تغطية جزئية، تغطية كثيفة، وتغطية المتكررة. بالإضافة إلى ذلك، سيؤدي النشر غير الصحيح لعقد المستشعر إلى فقد بعض العقد التي لن يكون لها أي جوار وسيتم عزلها وفصلها عن الشبكة. وبالتالي، فإن التغطية الأفضل هي مفتاح الأنظمة القوية ويمكن استغلالها لإطالة عمر هذا النوع من الشبكة بشكل كبير عن طريق تحويل العقد الزائدة إلى وضع السكون منخفض الطاقة. الهدف من عملنا في هذه الأطروحة هو استخدام خوارزمية مستعمرة النمل تسمى API-Apicals (Pachycondyla-apicalis) لتحسين تغطية الأهداف في هذه الشبكة مع ضمان القيود المختلفة مثل: التوصيلية في الشبكة وحياة الشبكة.

الكلمات المفتاحية: شبكة الاستشعار اللاسلكية، التغطية، التوصيل، خوارزمية مستعمرة النمل.

Abstract:

The coverage in Wireless Sensor Networks (WSN) is often considered to be a very important performance measure. It reflects how a given area is monitored, ie how each point in the surveillance zone is observed and followed by all nodes. This zone is often presented as a circle of radius r from where the node has a vision that depends on the radius of reception of its communication module r_s . The coverage of a WSN measures the degree of coverage of the space of interest by all the sensor nodes. This coverage can exist in three different types: partial coverage, dense coverage, and redundant coverage. In addition, a bad deployment of the sensor nodes will produce a loss of some nodes which will have no neighbors, and which will be isolated and disconnected from the network. Thus, better coverage is the key to robust systems and can be exploited to significantly extend the life of the WSN by switching redundant nodes to low energy sleep mode. The goal of our work is to use an ant colony algorithm called API-Apicals (*Pachycondyla-apicalis*) to optimize target coverage in the WSN while ensuring the various constraints of the WSN such as: connectivity in the WSN and the life of the WSN.

Key words: Wireless sensor networks, coverage, connectivity, ant colony algorithm.

Liste des figures

Figure 1.1 : un réseau Ad-hoc	03
Figure 1.2 : Exemple d'un réseau ad-hoc simple	04
Figure 1.3 : Architecture d'un réseau de capteurs	04
Figure 1.4 : Architecture d'un nœud de capteur.	05
Figure 1.5 : La transmission directe(Mono-saut)	06
Figure 1.6 : La transmission indirecte(Multi-saut)	07
Figure 1.7 : La pile protocolaire.	07
Figure 1.8 : Quelques domaines d'application	11
Figure 2.1 : La couverture de point.	15
Figure 2.2 : La couverture de zone.	16
Figure 2.3 : La couverture barrière.	16
Figure 2.4 : Diagramme de transition d'état du protocole PEAS.....	19
Figure 3.1 :Algorithme de colonies de fourmis.	25
Figure 3.2 : Les étapes utilisées par l'algorithme AS pour déterminer le chemin optimal	26
Figure 3.3 : Exploration des sites de nourriture par les fourmis <i>Pachycondyla-apicalis</i> ...	28
Figure 3.4 : Le comportement individuel de fourragement d'une fourmi. (n_s est le nombre de sites de chasse dans la mémoire de la fourmi et e_j le nombre d'échecs successifs rencontrés sur le site s_j)	30
Figure 4.1 : Une fenêtre du MATLAB.....	35
Figure 4.2 : un exemple de réseau entièrement couvert et connecté.....	36
Figure 4.3 : Fonctionnement du simulateur réalisé	38
Figure4.4 : Schéma du bloc fonctionnel du simulateur réalisé	39
Figure 4.5 : L'interface principale de notre simulateur.....	39
Figure 4.6 : Le déploiement aléatoire des nœuds de capteur	40
Figure 4.7 : La première expérimentation avec 4 nœuds de capteur et 4 cibles déployés de manière aléatoire	42
Figure 4.8 : La deuxième expérimentation avec 12 nœuds de capteur et 12 cibles déployés de manière aléatoire	42
Figure 4.9 : La troisième expérimentation avec 16 nœuds de capteur et 16 cibles déployés de manière aléatoire	43

Liste des figures

Figure 4.10 : La quatrième expérimentation avec 25 nœuds de capteur et 25 cibles déployés de manière aléatoire	44
Figure 4.11 : La cinquième expérimentation avec 50 nœuds de capteur et 50 cibles déployés de manière aléatoire	45
Figure 4.12 : Un organigramme représentant les nœuds actifs les nœuds désactivés.....	45

Liste des tableaux

Tableau 1.1 : comparaison entre capteurs et Ad-hoc	05
Tableau 4.1 : Les paramètres utilisés pour la simulation pour le RCSF.....	38
Tableau 4.2 : Les paramètres d'API	41

Table des abréviations

ACS	Area-based Collaborative Sleeping
ADC	Analog-to-digital Converter
API	Pachycondyla apicalis
AS	Ant System
CCSID	Connected Cover Set based on Identity of node
CGN	Carrier Grade Nat
DCovPDS	Distributed Coverage Preserving based on Dominating Set
EDMs	Electronic Management Document System
GPS	Global Position System
IOT	Internet of Object
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MMAS	Max-Min Ant System
PEAS	Probing Environnement and Adaptive Sleeping
PVC	problème du voyageur de commerce
QoS	Qualities of service
Rc	Rayon de communication
RCSF	réseaux de capteurs sans fil
TIC	Technologie de l'Information et de Télécommunication
TSP	Traveler Salesman Problem
WSN	Wireless Sensor Network

Table de matières

Table de matières

Introduction générale.....	01
<u>Chapitre 01 Généralités sur les réseaux de capteurs sans fil</u>	
1.1 Introduction	03
1.2 Réseaux ad-hoc	03
1.2.1 Définition d'un réseau Ad-hoc	03
1.2.2 Modélisation d'un réseau Ad-hoc	03
1.3 Les réseaux de capteurs sans fils.....	04
1.3.1 Définition d'un réseau de capteur sans fil... ..	04
1.3.2 Architecture d'un réseau de capteur	04
1.3.3 Les nœuds de capteurs.....	05
1.3.3.1 Définition d'un nœud de capteur.....	05
1.3.3.2 Architecture d'un nœud de capteur	05
1.3.4 La communication dans les RCSF	06
1.3.4.1 Transmission directe (Mono-saut)	06
1.3.4.2 Transmission indirecte (L'envoi par routage ad –hoc (Multi-saut))	07
1.3.5 La pile protocolaire	07
1.3.5.1 Les rôles des couches	08
1 La couche physique.....	08
2 La couche liaison de donnée	08
3 La couche réseau	08
4 La couche transport	08
5 La couche application.....	08
1.3.5.2 Les différents plans de gestion	09
1 plan de gestion de l'énergie	09
2 plans de gestion de la mobilité	09
3 plans de gestion des tâches.....	09
1.3.6 Les domaines d'application.....	10
1.3.6.1 Les applications militaires.....	10
1.3.6.2 Les applications médicales.....	10
1.3.6.3 Les applications commerciales	10
1.3.6.4 Les applications de sécurité.....	10
1.3.6.5 Les applications environnementales.....	10

Table de matières

1.3.6.6 Les applications domotiques	11
1.3.6.7 Les applications agricoles	11
1.3.7 Couverture	11
1.3.8 Connectivité	12
1.3.9 Durée de vie du réseau	12
1.4 Conclusion	12
<u>Chapitre 02 La couverture dans les réseaux de capteurs sans fil : État de l'art</u>	
2.1 Introduction	13
2.2 La définition de la couverture	13
2.3 Objectif de la couverture	13
2.4 Les travaux existents	13
2.4.1 La couverture de 1-couverture	14
2.4.2 La couverture de k-couverture	14
2.5 Les types de couverture	15
2.5.1 La couverture de points (cibles)	15
2.5.2 La couverture de zone	15
2.5.3 La couverture de frontière (barrière)	16
2.6 Le problème de couverture	17
2.7 Les paramètres de couverture dans les RCSF	17
2.8 Les critères liés au problème de couverture dans les RCSF	18
2.9 Classification des protocoles de la couverture de surface	18
2.9.1 Sondage de l'environnement et sommeil adaptatif	19
2.9.1.1 Protocole PEAS	19
2.9.2 Protocole affranchi de la découverte du voisinage	20
2.9.3 Protocole basé sur les ensembles dominants	20
2.9.4 Protocole basé sur les positions pour préserver la couverture	20
2.9.5 Solutions basées sur des méthodes géométriques	20
2.9.6 Heuristique distribuée et probabiliste de "clustering"	21
2.9.7 Couverture et connectivité intégrées	21
2.9.8 Algorithmes de plus faible ID et de plus grand Degré	21
2.9.9 Approche basée sur l'ordonnement d'activité des nœuds vidéo	22
2.10 Conclusion	22

Table de matières

Chapitre03 Généralités sur les Méta-heuristiques

3.1 Introduction	23
3.2 Définition d'heuristique	23
3.3 Définition de méta-heuristique	23
3.4 Les méthodes trajectoire	24
3.5 Les méthodes évolutionnistes (populaires)	24
3.5.1 Les algorithmes de colonies de fourmis	24
3.5.1.1 Algorithme système de fourmi : Ant System (AS)	25
3.5.1.2 Système de colonie de fourmis (ACS)	26
3.5.1.3 Max-Min Ant System (MMAS).....	26
3.5.1.4 Pachycondyla apicalis (API)	27
1 Principe de l'algorithme	27
2 Application de l'algorithme API	31
2.1 Opérateurs spécifiques à l'optimisation numérique	31
2.2 Optimisation de fonctions numériques	31
3.6 Conclusion	33

Chapitre 04 Implémentation

4.1 Introduction	34
4.2 L'environnement de simulation utilisé.....	34
4.3 La modélisation du problème utilisée	35
4.3.1 Le modèle mathématique utilisé	36
4.4 L'implémentation	37
4.4.1 Les paramètres de l'algorithme d'API utilisés	37
4.4.2 Les paramètres utilisés pour le RCSF pour la simulation	38
4.4.3 Fonctionnement du simulateur réalisé.....	38
4.4.4 Bloc fonctionnel du simulateur	39
4.4.5 La description du Simulateur et les résultats de simulation	39
4.5 Les paramètres d'API	41
4.6 Les expérimentations	41
4.7 Analyse des résultats	45
4.8 Conclusion	46
Conclusion générale	47
Bibliographie.....	48

Introduction générale :

Les réseaux de capteurs sans fil (RCSF) ont attiré l'attention de la communauté de la recherche et de l'industrie au cours des dernières années. Les efforts de recherche récents et le développement rapide des RCSF sont principalement motivés par leur application potentielle dans un large éventail de contextes, notamment les opérations militaires, la surveillance de l'environnement, les systèmes de surveillance, les soins de santé et la sécurité publique. Ces applications nécessitent le déploiement d'un certain nombre de capteurs pour couvrir une région d'intérêt donnée dans le domaine du réseau. Bien que les nœuds de capteurs puissent fonctionner de manière autonome, ils peuvent également travailler en collaboration pour surveiller les paramètres physiques d'un environnement. Les nœuds de capteurs peuvent détecter l'environnement, communiquer avec les nœuds voisins et, dans de nombreux cas, effectuer des calculs de base sur les données collectées. Ces fonctionnalités font du réseau un excellent choix parmi de nombreuses applications fonctionnant dans des environnements dangereux pour la présence humaine.

Le problème de couverture est l'un des problèmes fondamentaux des RCSF car il a un impact direct sur la consommation d'énergie des capteurs et la durée de vie du réseau. Le problème de couverture peut généralement indiquer comment surveiller efficacement le champ de réseau. Il existe plusieurs façons de classer les problèmes de couverture dans les RCSF. Les problèmes de couverture peuvent être classés, en fonction de la fréquence du moniteur de terrain du réseau, en problèmes de couverture continue ou en problèmes de couverture. Les problèmes de couverture continue peuvent être classés plus avant, en fonction de la région d'intérêt pour la surveillance, en trois types: couverture de zone, couverture de point et couverture de barrière. En outre, les problèmes de couverture peuvent être classés, en fonction du degré de couverture requis, en problèmes de 1-couverture ou en k-couverture.

D'autre part, les protocoles de couverture peuvent être classés en fonction des exigences de connectivité, en protocoles de couverture sensibles à la connectivité ou en protocoles de couverture non sensibles à la connectivité. Les protocoles de couverture peuvent également être classés en fonction de l'optimisation de la couverture, c'est-à-dire dans les protocoles de déploiement où la couverture est consciente, lorsque l'optimisation de la couverture intervient avant la phase de déploiement ou les protocoles de planification de la veille, lorsque l'optimisation de la couverture intervient après la phase de déploiement. Les protocoles de planification de sommeil peuvent également être classés, en fonction de la

topologie du réseau, en protocoles de planification de sommeil basés sur des clusters ou en protocoles de sommeil pour les réseaux plats.

Dans ce mémoire en adaptant un algorithme de colonies de fourmi appelé API (API-Apicalis) au problème de couverture simple (1-couverture) des cibles dans le RCSF afin d'améliorer sa durée de vie en sélectionnant un nombre minimum de nœuds de capteurs actifs parmi un grand nombre de nœuds de capteurs déployés de manière aléatoire. Les nœuds de capteurs sélectionnés doivent couvrir tout le nombre prédéfini de points cibles, de manière périodique ou continue. Simultanément, ces nœuds de capteurs actifs doivent maintenir la connectivité entre eux et la station de base (SB) de sorte que les précieuses données détectées puissent être transmises à la SB selon le plus court chemin.

La suite de ce mémoire est structurée en 04 chapitres :

- ✓ Nous présentons dans le premier chapitre des généralités sur les réseaux de capteurs sans fil;
- ✓ Le deuxième chapitre décrit un certain nombre de travaux de la couverture dans les réseaux capteurs sans fil. C'est quelque sorte un état de l'art;
- ✓ Dans le troisième chapitre nous présentons premièrement quelques Méta-heuristiques (les méthodes trajectoires et les méthodes évolutionnistes) puis nous détaillons l'algorithme de colonie de fourmi API (Pachycondyla-apicalis) adapté pour le problème de la couverture (1-couverture) de cibles du RCSF de notre travail.
- ✓ L'objectif du dernier chapitre décrit notre implémentation, la modélisation formelle utilisée pour adapté l'algorithme API à la couverture (1-couverture) dans le RCSF ainsi que l'interprétation des différents résultats trouvés.
- ✓ Nous achevons ce mémoire en donnant une conclusion générale, et quelques perspectives futures en vue d'améliorer et d'étendre ce travail.

Chapitre 01

Généralités sur les réseaux de capteurs sans fil

Chapitre 01

Généralités sur les réseaux de capteurs sans fil

1.1 Introduction :

Au cours des dernières décennies, nous avons assisté à une réduction de la taille des ordinateurs. Cette tendance a entraîné une miniaturisation de la nouvelle génération de réseaux informatiques de communication avec des défis importants. Les réseaux de capteurs sans fil (RCSF ou (WSN : Wireless Sensor Networks) en anglais) sont l'une des techniques utilisées pour résoudre les problèmes de cette nouvelle technologie de l'informatique embarquée et actuelle partout. Dans ce présent chapitre, on présentera un état de l'art sur les RCSF en décrivant leurs Définitions, architectures, connectivité, durée de vie, piles protocolaires et différents domaines d'applications. Les modèles de communication utilisés sont également étudiés.

1.2 Réseaux Ad-hoc :

1.2.1 Définition d'un réseau Ad-hoc :

Un réseau mobile Ad hoc (MANET) est un groupe d'unités mobiles, interconnectées par la technologie sans fil, qui constituent un réseau temporaire sans aucune infrastructure préexistante ou d'une gestion centralisée (voir la figure 1.1). Les unités mobiles (nœuds) interagissent et peuvent coopérer pour changer les services du réseau. [1].

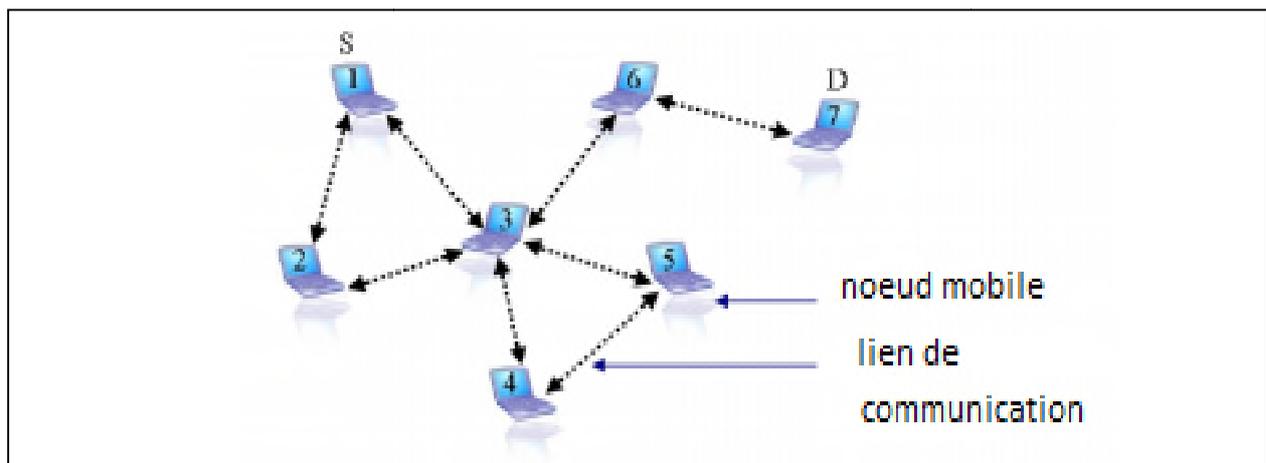


Figure 1.1 : un réseau Ad hoc.

1.2.2 Modélisation d'un réseau Ad-hoc :

Un réseau Ad hoc peut être représenté par un graphe non orienté $G = (V, E)$ où :

- V : représente l'ensemble des nœuds (les stations mobiles) du réseau à l'instant.

➤ E : modélise l'ensemble des connexions (arrêtes) existantes entre ces nœuds à un instant. Soient i et j deux nœuds de V , l'arc (i, j) existe si et seulement si i peut envoyer directement un paquet à j à l'instant, on dit alors que j est un nœud voisin de i [1].

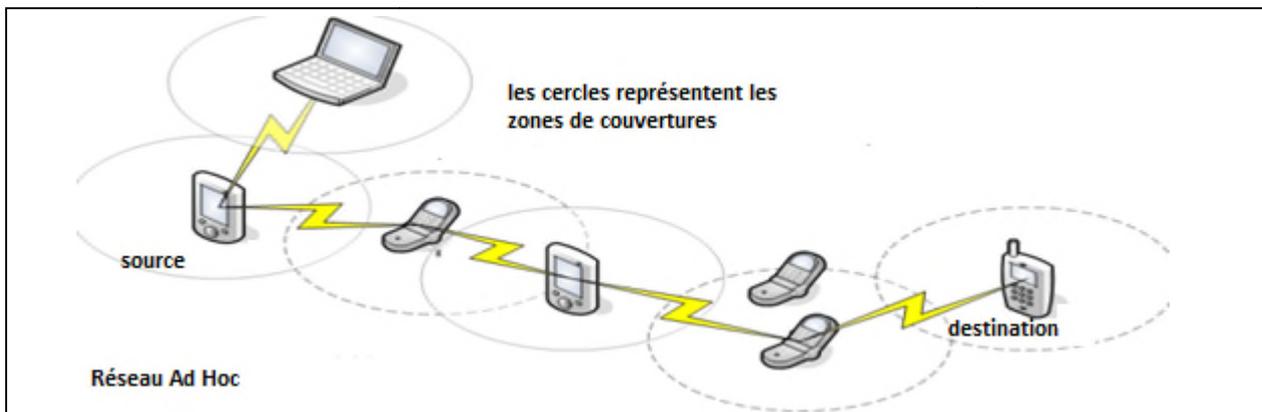


Figure 1.2 : Exemple d'un réseau Ad- hoc simple.

1.3 Les réseaux de capteurs sans fil :

1.3.1 Définition d'un réseau de capteur sans fil :

Le réseau de capteurs sans fil est un type spécial de réseau Ad hoc comprenant un grand nombre de nœuds de capteurs pouvant recevoir et transmettre des données environnementales indépendamment sans intervention humaine. La localisation de ces nœuds n'est pas nécessairement prédéterminée, elle peut être dispersée de manière aléatoire dans une zone géographique appelée "champ de captage" qui correspond à la zone d'intérêt du phénomène observé (par exemple, les capteurs de volcan projetés pour étudier les phénomènes volcaniques et leur évolution). Habituellement, le réseau a un nœud spécifique, qui représente la station de base (sink), qui représente le point intermédiaire (la passerelle) entre les différents nœuds du réseau avec l'utilisateur final [2].

1.3.2 Architecture d'un réseau de capteur :

Le réseau de capteurs est généralement constitué de plusieurs nœuds répartis dans la zone des capteurs.

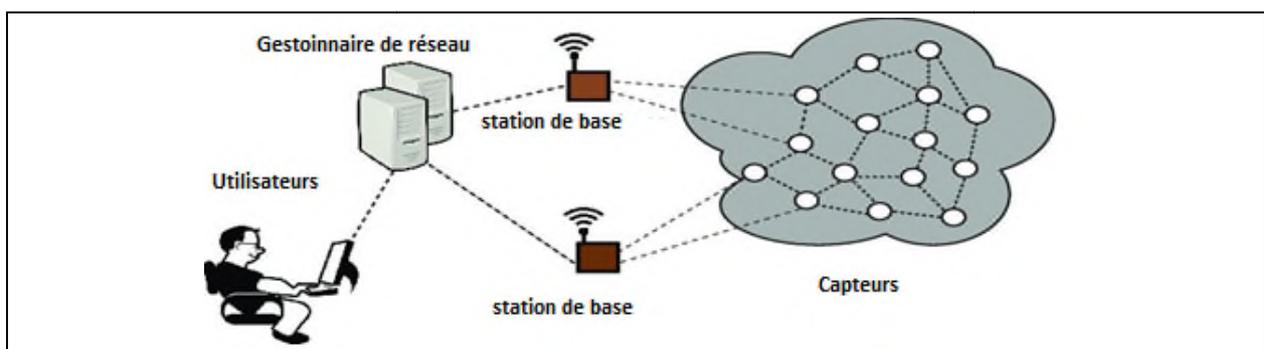


Figure 1.3 : Architecture d'un réseau de capteur.

Ce contrat concerne une ou plusieurs stations (sink) permettant l'interconnexion avec d'autres réseaux (internet, satellite ...) et récupération de données (voir la figure 1.3) [3].

1.3.3 Les nœuds de capteurs :

1.3.3.1 Définition d'un nœud de capteur :

Un nœud de capteur est un appareil capable de générer leur donnée environnement physique. Cet appareil initie le système de communication pour échanger des données et ainsi l'ensemble de ces nœuds capteur forment un réseau implicite. Le thème des "réseaux de capteurs" n'est pas d'inventer de nouveaux capteurs, mais de découvrir de nouveaux modèles de communication, leurs limites et leurs performances. La durée de vie des capteurs est limitée à la durée de vie de leur source d'alimentation (la batterie), le tableau suivant montre une petite comparaison entre les réseaux de capteurs et les réseaux Ad-hoc [4].

Les réseaux de capteurs	Ad-hoc
1. objectif ciblé	1. générique/communication
2. Nœuds en collaboration pour remplir un objectif	2. chaque nœud à son propre objectif
3. perte des données(many to one)	3. flot (any to any)
4. Très grand nombre de nœuds n'ayant pas tous un ID	4. notion d'ID
5. Energie (facteur déterminant)	5. Débit majeur
6. utilisation du broadcast	6. communication point à point

Tableau 1.1: comparaison entre les réseaux de capteurs et les réseaux Ad-hoc [4].

1.3.3.2 Architecture d'un nœud de capteur : Un nœud capteur est composé de quatre unités principales, qui sont présentées dans la figure 1.4 comme suit:

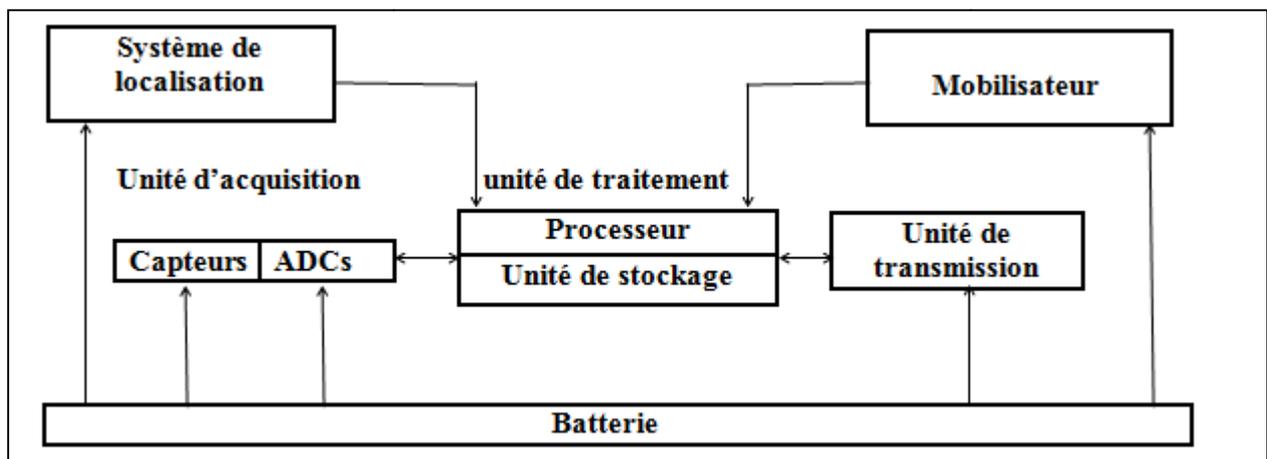


Figure 1.4 : Architecture d'un nœud de capteur.

➤ **L'unité d'acquisition (Sensing unit) :**

Se compose normalement de deux sous-unités: les convertisseurs analogiques-numériques ADC (analog-to-digital converter) et les capteurs. Les capteurs obtiennent des mesures numériques des normes environnementales et les convertissent en signaux analogiques. La sous-unité ADC convertit ces signaux analogiques en signaux numériques [5].

➤ **L'unité de traitement (Processing unit) :**

Cette unité contient deux sous-unités (un processeur et une mémoire) et se compose de deux interfaces: l'interface avec l'unité de transmission et l'autre avec l'unité d'acquisition. Elle contrôle les actions dans lesquelles le nœud coopère avec d'autres nœuds pour effectuer des tâches d'acquisition et stocke les données collectées [5].

➤ **Unité de transmission (Transceiver unit) :**

Il est responsable de toutes les communications via un support de connexion sans fil qui connecte le nœud au réseau [5].

➤ **Batterie (La source d'alimentation) (Power unit) :**

Alimente les unités citées précédemment. Les nœuds de capteur peuvent comprendre des composants supplémentaires tels que: les systèmes de localisation comme le GPS et le module du transport [5].

1.3.4 La communication dans les RCSF :

Dans ce type de réseau les informations récoltées ou réceptionnées par un nœud capteur dans ce dernier peuvent être transférées de deux manières :

1.3.4.1 Transmission directe (Mono-saut):

Chaque nœud est en contact étroit avec le sink et aucun nœud capteur intermédiaire ne peut interférer dans cette relation directe privilégiée [6].

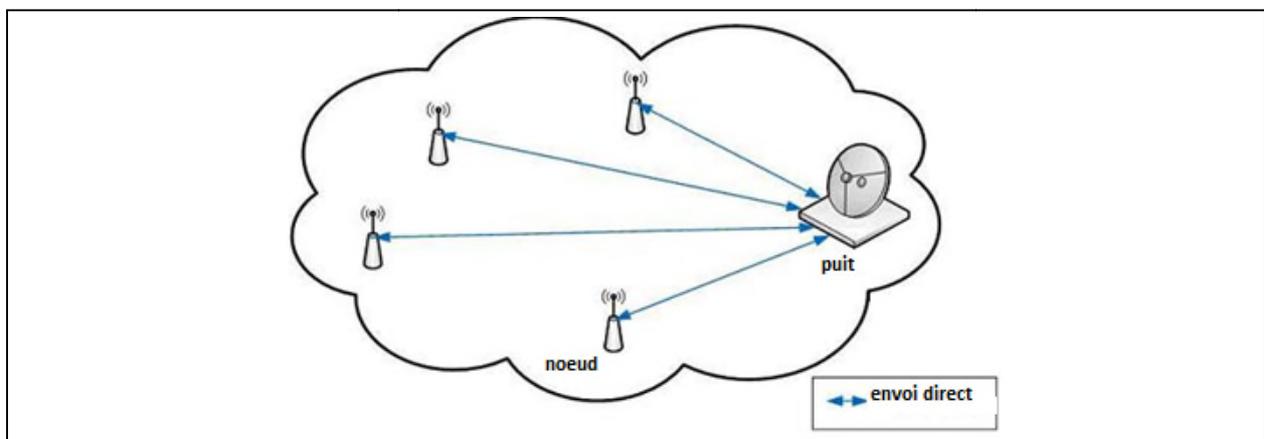


Figure 1.5 : La transmission directe (Mono-saut).

1.3.4.2 Transmission indirecte (L'envoi par routage Ad hoc (Multi-saut)) :

Lorsque les nœuds ne sont pas connectés à l'unité d'agrégation, la transmission directe n'est pas possible, il est donc nécessaire d'appliquer les règles de routage. Les réseaux de capteurs sont un type particulier de réseau dédié. Dans les réseaux personnalisés, il existe un seul type de périphérique, qui utilise le réseau et le forme. Les capteurs sont utilisés pour transmettre le trafic utile (informations à transférer) et le contrôle du trafic (informations à partager par les éléments du réseau pour détecter et maintenir les chemins sur le réseau) [6].

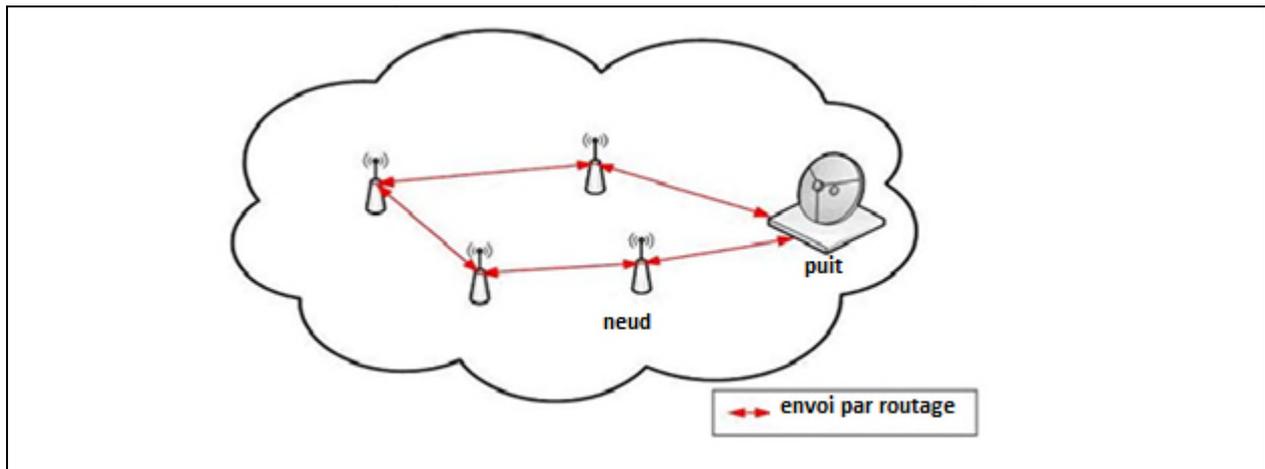


Figure 1.6 : La transmission indirecte (Multi-saut).

1.3.5 La pile protocolaire :

La figure 1.7 illustre la pile de protocoles utilisée par la station de base ainsi que par tous les autres nœuds capteur du réseau.

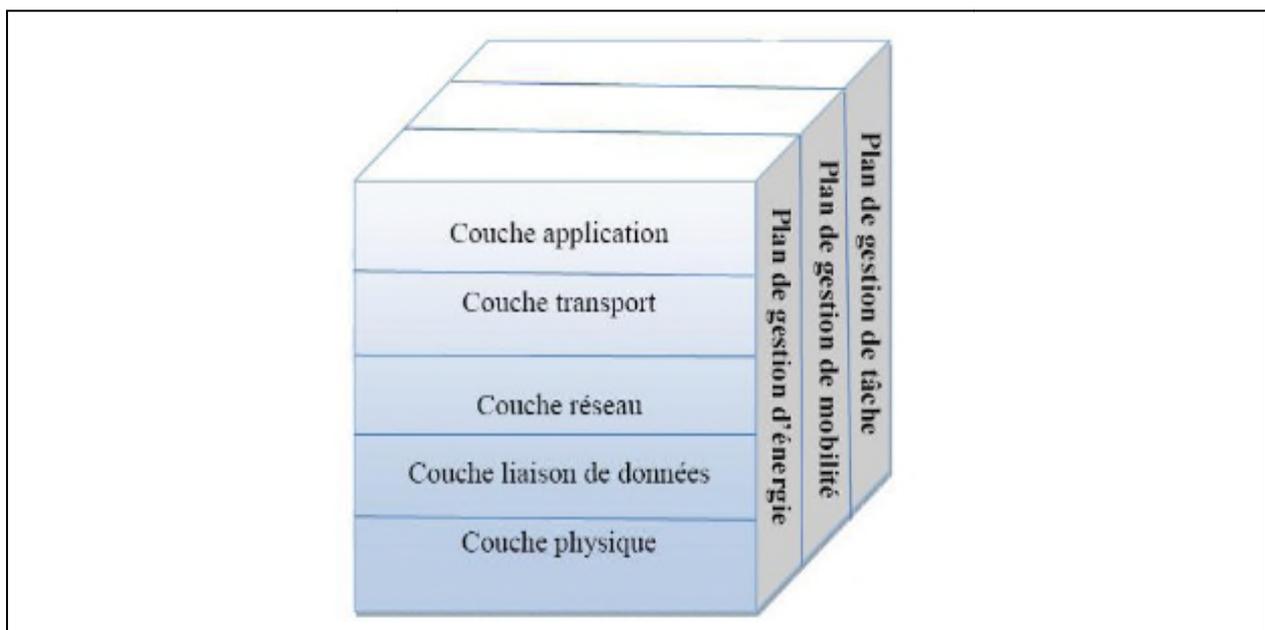


Figure 1.7 : La pile protocolaire.

Le package de protocole comprend la couche application, la couche de transport, la couche réseau, la couche liaison de données, la couche physique, le plan de gestion de l'alimentation, le plan de gestion de la mobilité et le plan de gestion des tâches. [7]

1.3.5.1 Les rôles des couches :

1- La couche physique :

La couche physique détermine les propriétés des dispositifs, les techniques de modulation, la détection et la conversion des signaux numériques, analogiques et optiques [8].

2- La couche liaison de données :

La couche liaison de données permet de contrôler le lien logique. Il permet également de déterminer comment les données sont transmises entre deux nœuds capteurs selon un saut ou plusieurs sauts. Ainsi, le protocole MAC (Medium Access Control) de cette couche de liaison permet de gérer l'accès au support physique en évitant les collisions qui surviennent souvent en raison de la compétitivité de l'accès à ce support physique. Cette couche est également responsable de la transmission multiple de données et du contrôle des erreurs de transmission. De plus, il fournit un lien de réseau point à point et multipoint dans le réseau [8].

3- La couche réseau :

Le rôle principal de la couche réseau est d'acheminer de manière fiable les données fournies par la couche de transport à la station de base, tout en essayant d'améliorer la consommation d'énergie induite par l'ensemble des nœuds capteurs participant à ce routage [8].

4-La couche transport :

La couche de transport est responsable d'un transfert de données et d'un contrôle de flux fiables. Pour bien exécuter ces fonctions, vous divisez les données de la couche application du nœud source en segments avant de les envoyer au réseau. Du côté du récepteur, ces données sont réorganisées à partir de la couche réseau et réassemblées avant d'être envoyées à la couche application. Cette couche peut également gérer les files d'attente de paquets avant de les déplacer vers la couche réseau [8].

5- La couche application :

La couche d'application est la couche supérieure. Il fournit une communication avec les applications. Ainsi, en fonction du type de nœud de capteur, différents types d'applications

peuvent être implémentés (mise en œuvre). Cette couche directement gérée par le programme utilisateur est le niveau le plus proche des utilisateurs. La couche d'application peut également gérer la collecte de données avant qu'elle ne soit déplacée vers la couche de transport. Les rôles de chacun des différents plans de gestion sont décrits dans les paragraphes suivants [8].

1.3.5.2 Les différents plans de gestion :

1- plan de gestion de l'énergie :

Le plan de la gestion de l'énergie contrôle l'utilisation de la batterie. Par exemple, après avoir reçu un message, la station de radio du nœud destinataire peut être désactivée afin que ce dernier puisse économiser son énergie. En outre, si le niveau de puissance d'un nœud capteur donné devient trop bas (généralement au-dessous d'un certain seuil), ce nœud peut envoyer une alerte à tous ses voisins pour les informer qu'ils ne peuvent plus participer à la communication dans le réseau, ce qui peut augmenter la tolérance aux pannes. Ainsi, la puissance restante de ce nœud peut être réservée à d'autres fonctions comme la capture [8].

2- plan gestion de la mobilité :

Le plan de gestion de la mobilité est utilisé pour détecter le mouvement d'un nœud de capteur particulier et l'enregistrer dans la zone d'intérêt. La rétraction est toujours réservée à l'utilisateur et le nœud peut suivre tous les nœuds voisins. Grâce aux informations sur l'état de ses voisins, les nœuds de capteurs peuvent mieux coordonner et gérer leur consommation d'énergie afin de mener à bien leurs différentes tâches [8].

3- Plan gestion des tâches :

Enfin, le plan de gestion des tâches permet de planifier des diverses tâches de collecte de données dans une zone de contrôle particulière. Dans certaines topologies de déploiement denses où les champs de capture sont souvent répétitifs, il n'est pas nécessaire que tous les points de détection d'une zone particulière effectuent une tâche spécifique en même temps. Ainsi, le nœud capteur peut conserver les fonctions de capture lorsque d'autres nœuds entrent en mode hors tension. D'après la description des différentes fonctions de gestion ci-dessus, nous constatons qu'elles sont nécessaires à la coordination entre les différents nœuds, à la gestion de leur mobilité et à l'amélioration de la consommation d'énergie dans le réseau. Par conséquent, avec une bonne utilisation intelligente de ces fonctions et en fonction du type d'application, les performances de RCSF peuvent être améliorées [8].

1.3.6 Les domaines d'application :

Les réseaux de capteurs sans fil peuvent être utiles dans de nombreux domaines d'application [9]. Dans cette section on présente la description de quelques domaines d'application.

1.3.6.1 Les applications militaires :

Les réseaux de capteurs étaient utilisés à des fins militaires. En effet, déploiement rapide, faible coût, autorégulation et tolérance aux pannes font de ce type de réseau un outil précieux dans un tel domaine. Le réseau de capteurs peut être déployé dans un endroit stratégique ou difficile d'accès pour surveiller toutes les activités de l'ennemi ou pour analyser le terrain avant d'envoyer des troupes (détection par rayonnement ou chimique) [9].

1.3.6.2 Les applications médicales :

Surveiller les constantes physiologiques du patient sera possible grâce à des nœuds de capteur précis pouvant être implantés sous la peau. Utiliser des réseaux de capteurs dans le domaine de la médecine peut permettre une surveillance patiente, collecter les meilleures informations et ainsi faciliter le diagnostic de certaines maladies et intervention rapide si les mesures sont fabriquées par des capteurs anormaux [9].

1.3.6.3 Les applications commerciales :

Des nœuds capteurs peuvent être intégrés aux opérations de stockage et de livraison. Pour les entreprises manufacturières, les réseaux de capteurs pourront suivre le processus de production des matières premières en produit final. Grâce à l'utilisation de ces derniers, les entreprises peuvent fournir un service de meilleure qualité [9].

1.3.6.4 Les applications de sécurité :

Les réseaux de capteurs de sécurité peuvent être réduits largement des charges financières consacrées à l'assurance des bâtiments, protéger les humains tout en assurant de meilleurs résultats. Par exemple, la modification de la structure du bâtiment après un séisme ou un vieillissement, ils peuvent être détectés par des capteurs encastrés dans des murs ou du béton [9].

1.3.6.5 Les applications environnementales :

Les nœuds capteurs dans une forêt peuvent indiquer la possibilité d'un incendie, Ce qui permettra une meilleure efficacité dans la lutte contre les incendies de forêt. Dans les champs agricoles, le capteur peut être planté avec des semences. Dans des endroits d'installations industrielles, centrales nucléaires ou pétroliers, des nœuds capteurs peuvent

être déployés pour détecter des fuites de produits toxiques (gaz, produits chimiques, etc.) et alerter les utilisateurs ont suffisamment de temps pour permettre une intervention efficace [9].

1.3.6.6 Les applications domotiques :

Avec le développement de la technologie, des nœuds capteurs peuvent être intégrés dans des appareils tels que des aspirateurs, des fours à micro-ondes, des réfrigérateurs, des enregistreurs vidéo, etc. [10]. Ces capteurs intégrés peuvent interagir les uns avec les autres et avec un réseau externe sur internet pour permettre à l'utilisateur de contrôler ses appareils ménagers localement ou à distance.

1.3.6.7 Les applications agricoles :

Les nœuds capteurs peuvent être plantés avec des graines. Ainsi, les zones sèches seront facilement identifiées et l'irrigation sera plus efficace et économique [11].

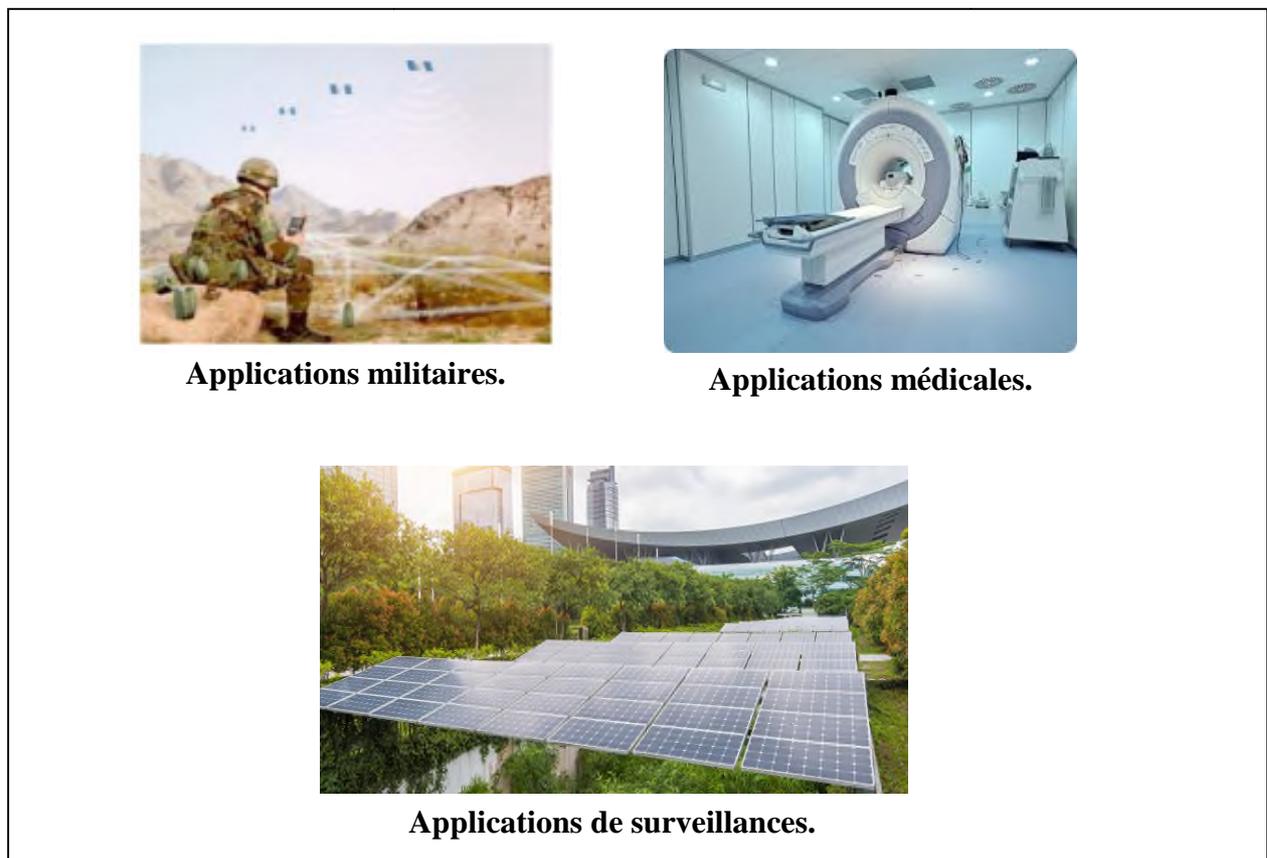


Figure 1.8 Quelques domaines d'application.

1.3.7 Couverture :

La fonction principale d'un réseau de capteurs est d'observer une zone physique sous les diverses contraintes et limitations du contrat de capteurs. Les limites incluent: l'énergie, la capacité de calcul, la mémoire, la communication, la bande passante, taux d'échec élevé, etc.

Cette surveillance doit répondre à un certain nombre d'exigences, notamment la durée de vie du réseau et la qualité de la surveillance [7].

1.3.8 Connectivité:

Il faut que la connectivité dans les RCSF doive être assurée juste après que la phase de déploiement est terminée dans la zone d'intérêt, les nœuds doivent former un réseau connecté afin de transférer les informations capturées par les nœuds source vers la station de base. Selon le type d'architectures utilisées, tous les nœuds de réseau ou une partie de ce groupe doivent être connectés en permanence une fois qu'une source de données est transférée à la station de base [8].

1.3.9 Durée de vie du réseau :

La durée de vie est une mesure importante de l'évaluation du rendement du RCSF. Il existe plusieurs définitions pour ce paramètre. Cependant, notez que sa définition n'est pas toujours triviale. En fait, dans [12], les auteurs présentent la durée de vie d'un RCSF en ce qui concerne l'application et les protocoles utilisés. Dans [13], la conception de la durée de vie du RCSF est une mesure étroitement liée à la vie des nœuds du réseau. Pour les auteurs de [14], il définit la durée de vie comme une autre mesure de l'accès aux nœuds du réseau. Hu et Li suggèrent que le terme CGN est défini comme étant lié à la QoS [15], et d'autres chercheurs, mesure liée à la distribution spatiale des nœuds du réseau [16].

1.4 Conclusion :

Les réseaux de capteurs sans fil revêtent une grande importance et constituent une nouvelle étape dans le développement des TIC. Cette nouvelle technologie suscite un intérêt croissant en raison de la diversité de ces applications: santé, environnement, industrie et même sport. Dans ce chapitre, nous avons présenté les réseaux de capteurs sans fil, leurs propres structures de communication et une gamme de protocoles de capteurs et leurs diverses applications. Cependant, nous avons constaté que de nombreux facteurs et contraintes compliquent la gestion de ce type de réseau. En fait, les économies d'énergie et la l'amélioration de la durée de vie sont des tâches clés dans le RCSF. En outre, pour surveiller le nœud capteur de réseau à grande échelle est étendu à une couverture maximale. À cette fin, plusieurs stratégies ont été proposées dans la littérature pour programmer l'activité des nœuds capteurs déployés dans une zone d'importance pour assurer la couverture de la zone. Parmi ces contraintes, nous concentrons sur la couverture des cibles (zones). Le chapitre suivant présente un état de l'art sur la couverture dans les RCSF.

Chapitre 02

La couverture dans les réseaux
de capteurs sans fil : État de
l'art

Chapitre 02

La couverture dans les réseaux de capteurs sans fil : État de l'art

2.1 Introduction :

L'un des défis majeurs des réseaux de capteurs est la couverture. Cette dernière permet la surveillance de phénomène ou d'événements continus ou effectif. Cela peut se produire dans une zone géographique. La couverture dans RCSF joue un rôle plus important dans le succès des applications du réseau de capteurs, en fait l'efficacité relative de RCSF à un taux. Une couverture spécifique, par exemple des réseaux de capteurs dans le domaine militaire, doit être assurée une couverture complète de la zone d'intérêt.

2.2 La définition de la couverture :

La couverture de surface peut être définie comme la capacité où les nœuds capteurs peuvent surveiller une région. Les nœuds capteurs ne doivent pas uniquement couvrir la surface et les données communiquez également les uns avec les autres pour intégrer, comparer des informations, afin de les transmettre au sink via des communications multi-saut. Pour exploiter les données collectées, le réseau doit être fortement connecté, c'est-à-dire qu'il est nécessaire que tous les nœuds trouvent un chemin à la station de base pour transférer les fruits de l'observation. La couverture de la surface fait l'objet de nombreux thèmes de recherche et dépend aussi des travaux en cours de la théorie des graphes. [7]

2.3 Objectif de la couverture :

- Analyser une architecture de réseaux de capteurs sans fils
- Analyser les protocoles de transport dans les RCSF
- Développer des applications pour les réseaux de capteurs sans fils
- Analyser les vulnérabilités et les solutions de sécurité pour RCSF
- Analyser les vulnérabilités et les solutions de tolérance aux pannes dans les RCSF

2.4 Les travaux existants :

La couverture de zone peut prendre des formes diverses selon la nature d'application par exemple, dans des applications moins sensibles telles que la surveillance.

Pour les champs agricoles, les protocoles de couverture peuvent être conçus de manière à ce que chaque point d'intérêt soit surveillé par un nœud de capteur et, dans certains cas non garanti pas nécessairement une couverture complète de la zone d'intérêt. Dans ce cas, on parle de 1-couverture ou une simple couverture. Cependant, dans les applications sensibles telles que les applications militaire ou de la sécurité, il est nécessaire d'assurer la couverture de chaque point du champ d'intérêt est constitué de plusieurs nœuds capteurs permettant la tolérance aux pannes. Dans ce cas, on parle de la couverture de k ou de plusieurs couvertures. Dans cette section, nous présentons les travaux en cours sur les deux modèles couverture: 1-Couverture et k - couverture.

2.4.1 La couverture de 1-couverture :

Les auteurs dans [17] ont proposé un algorithme de couverture de zone appelé PEAS (sondage environnement et veille adaptative) pour les réseaux de capteurs asynchrones. D'abord, tous les nœuds de capteurs sont dans un état de veille et après un certain temps, s'il nœud de capteur veut passer en mode actif (éveillé), il envoie un message à l'un de ses voisins. Ce dernier évalue la distance du nœud de capteur émetteur en fonction de la force du signal reçu ou délai de la transmission. Si cette distance est inférieure à une certaine distance seuil P , il demande toujours à ses voisins de rester dans un état de veille près du nœud de capteur émetteur et couvre donc sa zone de détection. Sinon, si le nœud de capteur émetteur ne reçoit aucun message du voisin, il décide donc de passer en mode actif, maintenu en attente jusqu'à épuisement de la batterie. Dans PEAS, le nœud reste actif jusqu'à ce qu'il tombe en panne ou qu'il manque de l'énergie de la batterie. Ensuite, il remplace les nœuds en mode veille si les nœuds ont échoué, rendant PEAS sans erreurs tolérante. Cette technique ne peut pas être souhaitable car l'intensité des nœuds actifs se dégradera avec le temps. De plus, dans l'environnement où la principale cause de défaillance du nœud capteur est l'épuisement de leur batterie, il est recommandé d'équilibrer la consommation d'énergie entre les nœuds du réseau. Pour cela, le contrat actif doit être choisi périodiquement et sur la base de plusieurs facteurs tels que l'énergie et la densité du réseau. De plus, PEAS ne garantit pas la couverture totale de la zone d'intérêt, sauf si une relation étroite est établie entre la distance et le seuil de détection P et R_s (rayon de la couverture).

2.4.2 La couverture de k -couverture :

La zone est appelée k -couverte si n'importe quel point p peut être surveillé par au moins un nœud actif. Nous ferons référence à cette définition en parlant de k - couverture.

La définition la plus stricte est que k-couverture se compose de k sous-ensembles distincts de nœuds actifs, également appelés couches ou niveaux d'activité.

2.5 Les types de couverture :

Selon la nature de l'objet à surveiller, la couverture est divisée en trois types: La couverture de zone, la couverture de points (les cibles), la couverture de frontières (barrières).

2.5.1 La couverture de points (cibles):

L'objectif principal dans ce type de couverture est de couvrir un ensemble de points spécifiques, dont la position géographique est connue. La figure 2.1 montre un exemple d'un ensemble de capteurs déployés aléatoire pour couvrir un ensemble de points (petits carrés) où les nœuds noirs reliés forment l'ensemble des capteurs actifs.

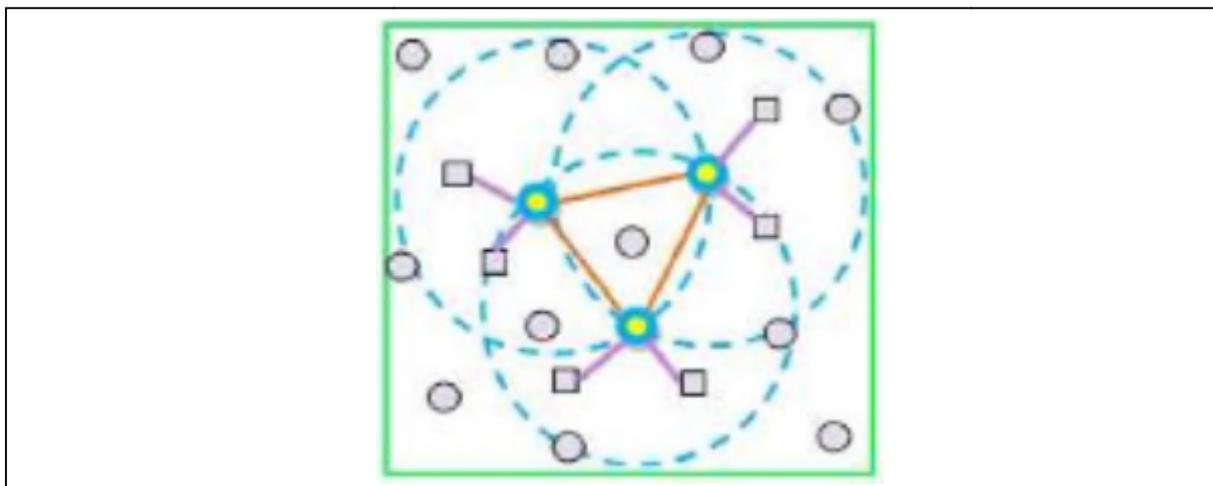


Figure 2.1 La couverture de points.

Selon la nature de l'objet à surveiller, le type de cible à surveiller est divisé en deux types : les cibles fixes et les cibles mobiles.

2.5.2 La couverture de zones :

S'appelle aussi la couverture de surface où l'objectif principal du réseau est de surveiller une région donnée [18,19], c.à.d. la zone d'intérêt est représentée comme une région où les points de cette région sont couverts par au moins un nœud capteur. La figure 2.2 montre un exemple où les nœuds capteurs sont déployés aléatoirement sur une zone rectangulaire afin d'assurer la couverture de cette dernière. Les cercles représentent les rayons de surveillance des nœuds capteurs, tout point de la zone est couvert par au moins un nœud capteur.

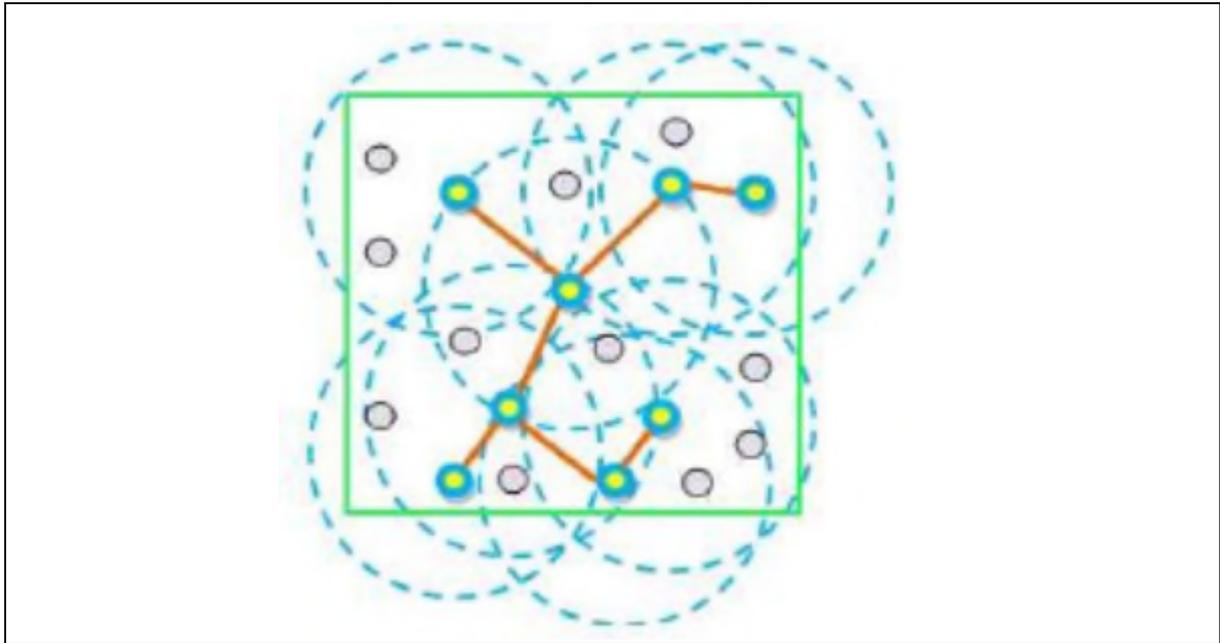


Figure 2.2 : La couverture de zones.

2.5.3 La couverture de frontières (barrières):

La limite commune d'application pour le RCSF consiste à surveiller la zone où les nœuds capteurs sont déployés dans une zone longue et étroite située sur une longue distance de la frontière, afin de détecter une intrusion non autorisée. Ce type d'application, également appelé boîtier permettant d'assurer la couverture à travers la barrière. Ce dernier, attire l'attention de plusieurs chercheurs dans ces dernières années en raison afin de les améliorer dans le but de l'obligation de protéger les informations sensibles, des installations militaires et des frontières des états. La figure 2.3 montre un exemple de ce type de couverture

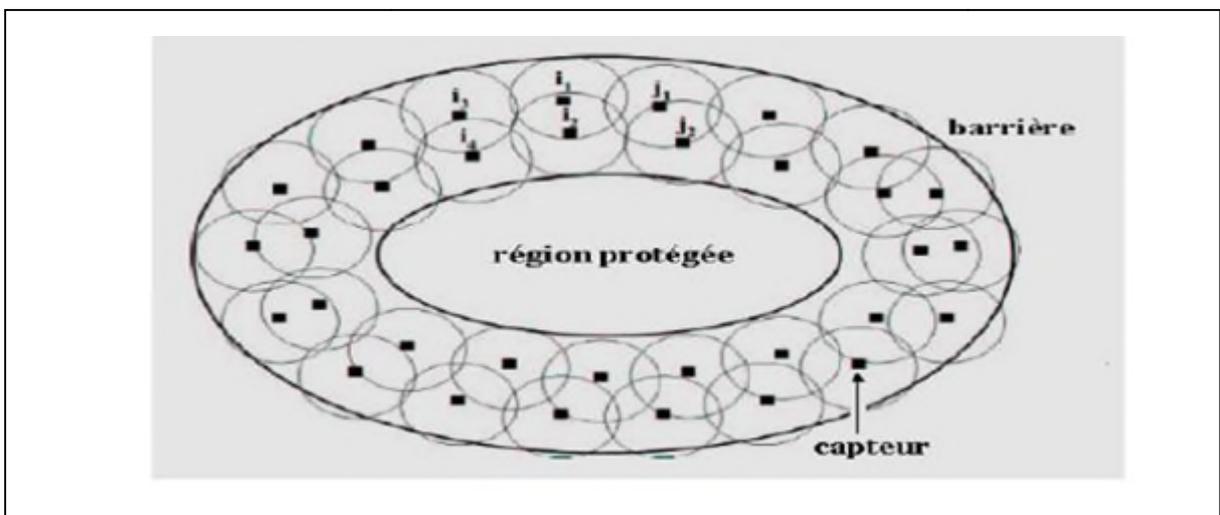


Figure 2.3 : La couverture de barrières.

2.6 Le problème de couverture :

Le problème de la couverture est centré sur une question fondamentale: "Comment un nombre de nœuds capteurs surveille-t-il un domaine d'intérêt particulier?" [19,20]. La couverture est une mesure de la qualité de service (QoS) de la fonction de surveillance fournie par le réseau de capteurs [21,22]. L'objectif est de surveiller tous les points physiques de la région d'intérêt dont le rayon de capture de nœuds regroupe au moins un nœud capteur [22].

2.7 Les paramètres de couverture dans les RCSF :

La couverture dans le RCSF doit prendre en compte l'amélioration d'un ou plusieurs paramètres considérés comme des métriques essentielles applicables :

➤ Nombre de nœuds capteur :

Afin d'équilibrer la charge de la couverture de la surface surveillée dans le RCSF et d'augmenter la durée de vie de ce dernier on doit basculer une partie de ses nœuds de capteur en état de veille.

➤ Qualité de la surveillance :

Le contrôle de la qualité est une mesure de la performance de la surveillance dans ce type de réseau [23]. Il est scanné en moyenne par une cible avant d'être détecté par un nœud capteur. Selon cette définition, le nœud capteur de qualité à une surveillance de haute qualité, peut détecter le mouvement de la cible à une distance plus courte. La qualité du contrôle n'est pas une simple couverture, mais également des facteurs tels que la répartition géographique des nœuds capteur.

➤ Connectivité du réseau :

Si les données collectées doivent traverser un chemin à sauts multiples avant d'atteindre la destination (telle qu'une station de base), il est important de garder les nœuds connectés. Certaines applications peuvent même configurer le réseau pour un certain degré de connectivité.

➤ La durée de vie :

C'est l'intervalle entre le déploiement du réseau et le moment où le premier trou de couverture apparaît [24]. Une autre définition est le temps écoulé jusqu'à ce que le nombre de nœuds tombe en panne ou que l'énergie résiduelle moyenne tombe en dessous du seuil prédéterminé [25,26]. En outre, le réseau est considéré comme fonctionnel lorsque le taux de couverture est supérieur à un certain seuil [27] ou tant que la station de base reçoit des rapports de surveillance.

➤ **Tolérance aux pannes :**

C'est la capacité de résister à un échec imprévu. Par exemple, les nœuds dispersés dans une application militaire peuvent être abattus avant l'épuisement de leur source d'alimentation. Une application est fiable doit maintenir toujours son fonctionnement même si un groupe de nœuds est défaillant.

2.8 Les critères liés au problème de couverture dans les RCSF :

Dans la littérature, les applications et les protocoles traitant du problème de couverture varient, en fonction de certains critères, ces paramètres peuvent être considérés comme des hypothèses permettant de résoudre le problème de couverture dans RCSF.

➤ **Structure du réseau :**

Un réseau de capteurs peut utiliser une architecture hiérarchique ou plate. Dans l'architecture plate tous les nœuds capteurs ont les mêmes rôles et fonctions [28]. Par contre, dans un réseau hiérarchique, les nœuds capteurs sont organisés en sous-groupes où les chefs de groupe ont un rôle plus important que les autres. Par exemple, dans les réseaux conçus pour détecter et suivre des objets, certains nœuds capteurs peuvent être désignés comme centres de fusion: ils collectent des données à partir d'autres nœuds capteurs adjacentes envoient un rapport à la station de base.

➤ **Modèle d'un nœud capteur :**

Le nœud capteur peut détecter un objet ou un événement si l'objet se trouve dans sa propre zone d'observation, appelée modèle de détection déterministe. Cependant, certains travaux utilisent un modèle de capture de probabilité [29, 30, 31], dans lequel la probabilité de détecter un objet égale à une fonction de la distance qui sépare cet objet du nœud capteur.

➤ **Degré de couverture :**

Dépendant de l'exigence des applications en termes de précision des données et de fiabilité, le problème de couverture peut être divisé en deux types : couverture simple (1-couverture) et couverture multiple (k-couverture). Cette dernière peut être employée pour accroître la précision des relevés ou diminuer les risques de fausses alertes.

2.9 Classification des protocoles de la couverture de surface :

La plupart des protocoles de couverture de surface utilisent un outil pour évaluer la couverture: tout nœud doit décider en premier lieu l'état de son activité pour évaluer la couverture fournie par les voisins de communication. Un nœud capteur ne peut être en état de veille que si sa zone est contrôlée par eux.

2.9.1 Sondage de l'environnement et sommeil adaptatif :

2.9.1.1 Protocole PEAS :

Dans [32], l'auteur suggère PEAS (Probing Environment and Adaptive Sleeping), un algorithme de couverture de zone conçu pour les réseaux de capteurs denses et asynchrones, utilisé dans un environnement où le déploiement planifié est risqué ou impossible en raison de sites hostiles où l'intervention humaine n'est pas possible (zones inaccessibles) ou lié à un grand nombre de nœuds de capteurs déployés dans une situation où la taille de la zone à surveiller est très grande. Ce protocole est aléatoire et sa distribution est divisée en deux phases:

➤ *Sondage de l'environnement :*

Pour passer du mode veille au mode éveillé le nœud capteur doit sonder son environnement par l'envoi d'un message de sondage à ses voisins, si ses derniers se trouvent tout près de lui et ils couvrent sa zone de détection alors il lui demande de rester dans l'état veille sinon si aucun autre nœud voisin à une distance d autour de lui n'est actif il décide alors de se mettre en mode actif jusqu'à l'épuisement de son énergie.

➤ *Sommeil adaptatif :*

Le nœud du capteur reste en état de veille jusqu'à ce qu'un nœud actif tombe en panne ou que sa batterie soit épuisée. Le remplacement ensuite de ce nœud défaillant est effectué s'il est nécessaire.

➤ *Description du protocole PEAS :*

PEAS est un protocole local simple utilisé pour les réseaux asynchrones. Le diagramme suivant montre les différentes transformations d'état de PEAS.

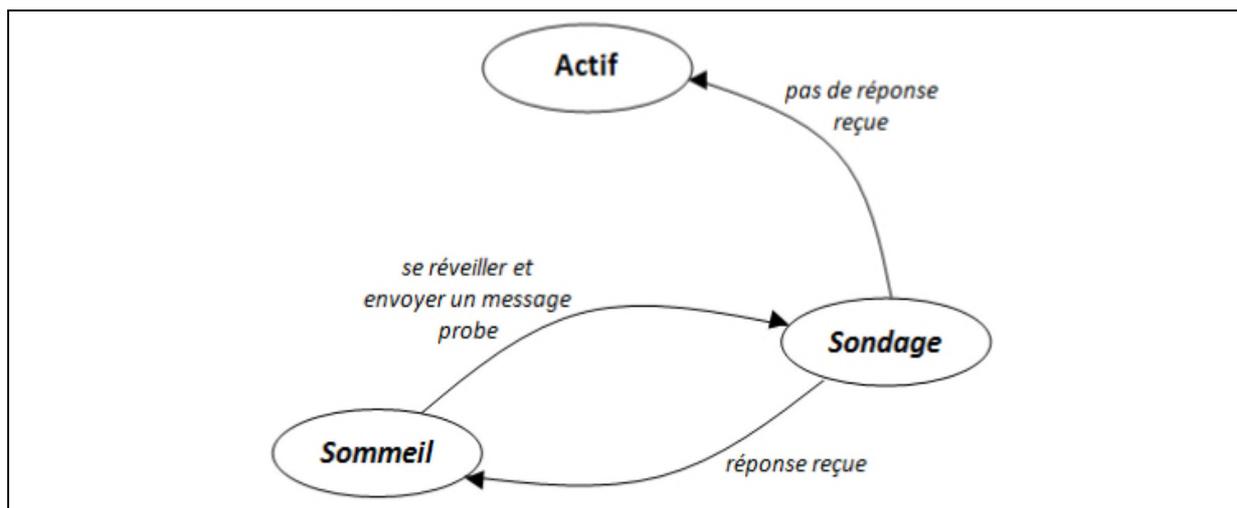


Figure2.4 Diagramme de transition d'état du protocole PEAS.

Au démarrage, tous les nœuds basculent en état de veille et après l'expiration de la durée de veille d'un nœud, alors il bascule en état actif (éveillé) afin de transmettre un

paquet de sondage appelé PROBE. Ce dernier peut être réceptionné par tous les nœuds actifs ayant le même rayon de communication R_c où chacun de ces nœuds capteurs vérifie si la distance de l'émetteur est inférieure à une distance d utilisée par chaque nœud. Ce calcul peut être effectué à partir de la force du signal reçu ou de délais de transmission. Un nœud voisin répond l'émetteur par l'envoi d'un message appelé REPLAY, et il passe automatiquement en état de veille. Sinon, l'émetteur reste en état actif (éveillé) s'il ne reçoit aucun message jusqu'à l'épuisement de sa source d'alimentation.

2.9.2 Protocole affranchi de la découverte du voisinage : Protocole de Gallais et al

Dans [33] Gallais et al, Proposent un protocole distribué et spécifique, dans lequel chaque nœud capteur est doté d'un système de GPS utilisé afin de localiser sa position. Il est utilisé dans des réseaux de capteurs synchrones où les nœuds capteur utilisent les mêmes rayons de détection et de communication dont la portée est la même. Les messages d'échange utilisés par les nœuds contiennent leur position et leurs états dans le but d'assurer l'évaluation de la connectivité de l'ensemble des nœuds voisins.

2.9.3 Protocole basé sur les ensembles dominants :

Dans [34], l'auteur a proposé un protocole de distribution pour résoudre le problème de couverture dans RCSF, nommé DCovPDS (Distributed Coverage Preservingbased on Dominating Set). Le protocole proposé permet la construction des EDMs de manière distribuée. En fait, chaque nœud de capteur exécute le protocole indépendamment des autres nœuds pour déterminer son statut: est-il dominant ou non (dominé)?

DCovPDS divise la durée de vie du réseau en périodes d'activités successives de même durée, chacune composée des deux étapes suivantes: Phase de décision, Phase de capteur.

2.9.4 Protocole basé sur les positions pour préserver la couverture :

Dans [35], l'auteur a proposé un protocole appelé CCSID (Connected Cover Set based on Identité of node) afin de résoudre le problème de couverture dans les RCSF qui est représenté comme un problème de recouvrement minimal d'un graphe où l'objectif de ce dernier est de trouver le plus petit ensemble de nœuds connectés assurant une couverture maximale.

2.9.5 Solutions basées sur des méthodes géométriques :

Une autre approche utilisant la mobilité [36], a été utilisée pour augmenter la couverture de la zone de contrôle.

Le but de cette approche est de restaurer les points de capteur de manière optimale pendant le fonctionnement du réseau ou après le déploiement initial du réseau. En pratique, le déploiement aléatoire initial ne garantit souvent pas une couverture complète et peut

entraîner une disparité importante dans la distribution entre les différentes parties de la zone de contrôle. De plus, des phénomènes environnementaux ou des changements topologiques (tels que l'ajout ou la perte de nœuds) peuvent survenir pendant le fonctionnement du réseau et ainsi rendre le réseau non amorçable. D'où la nécessité de rétablir le contact dans des parties de la zone de contrôle où la couverture est faible

2.9.6 Heuristique distribuée et probabiliste de "clustering"

Dans [37], suggèrent une utilisation distribuée et utile le "montage" détermine le nombre de NC en fonction du délai k (en nombre de sauts) de bout en bout attendu. Les centres sont identifiés par un processus électoral basé sur le « Groupe dominant indépendant » au k -saut. Un k -saut est défini par un groupe distinct constitué d'un contrat fort appelé "dominante" (dont la mission est de relayer données NC) et qu'aucun des nœuds dominants n'est adjacent à un autre nœud dominant. L'autre contrat, appelé "Dominator", doit pouvoir atteindre au moins un nœud coussins avec un maximum de k sauts. L'élection du nœud dominant est généralement basés sur des mesures telles que le degré de connectivité [38].

2.9.7 Couverture et connectivité intégrées :

Dans [39], suggèrent de jouer à des densités de contrat pour assurer le contact et la couverture de la zone de contrôle. Ils montrent que la topologie est optimale si la densité du contrat est également optimisée.

2.9.8 Algorithmes de plus faible ID et de plus grand Degré

Dans [40], les auteurs ont proposé un autre algorithme, l'identificateur le plus bas", pour construire des blocs dans lesquels chaque nœud était déclaré tête ou dépendant de sa connaissance et de celle de ses voisins, ainsi que de ses voisins. Dans l'algorithme "Smaller ID", le nœud peut contenir quatre états: normal, en-tête de groupe, membre ou passerelle. Initialement, tous les nœuds ont un état de nœud normal. Si le nœud u contient le plus petit identifiant dans son voisinage 1. Il s'annoncera comme l'en-tête de groupe et ses premiers voisins, dont la connaissance est supérieure au groupe et devient membre. Sinon, il attendra que ses voisins annoncent leur statut. Donc, si l'un d'entre eux se déclare l'en-tête du groupe, le nœud déclare l'état de son fils. Si tous les voisins de nœud ont l'ID est plus petit que celui qui contient un état membre de nœud et le nœud u déclarera l'en-tête de cluster. Une fois que tous les nœuds contiennent une condition de membre ou en-tête, si le nœud contient l'un de ses voisins avec plus d'en-tête de groupe, il se déclarera une passerelle.

2.9.9 Approche basée sur ordonnancement d'activité des nœuds vidéo

➤ Ordonnancement de l'activité des nœuds vidéo

Dans [41], les auteurs proposent une approche de planification d'activité pour couvrir le nœud vidéo fonctionnant en deux étapes: (1) initialisation au cours de laquelle chaque nœud construit ses propres groupes de couverture, (2) planification de l'activité et le nœud décidera d'être actif ou non.

2.10 Conclusion

La couverture est l'une des mesures les plus importantes pour l'évaluation de la produite par le réseau de capteurs dans une zone géographique. Une zone dite couverte, si tous les points avec au moins un capteur sont observés. Dans ce chapitre nous avons abordé le problème de la couverture de surface puis décrit quelques travaux existant.

Chapitre 03

Généralités sur les Méta- heuristiques

Chapitre 03

Généralités sur les Méta-heuristiques

3.1 Introduction :

La majorité des problèmes d'extraction de connaissances peuvent s'exprimer comme des problèmes d'optimisation combinatoire. Or, de nombreux problèmes d'optimisation combinatoire sont NP difficiles et ne pourront donc pas être résolus de manière exacte dans un temps «raisonnable» puisque la capacité de calcul des machines évolue linéairement alors que le temps nécessaire à la résolution de ces problèmes évolue exponentiellement. Lorsqu'on s'attaque à des problèmes réels, il faut se résoudre à un compromis entre la qualité des solutions obtenues et le temps de calcul utilisé. L'objectif principal de ce chapitre est de présenter l'algorithme de colonies de fourmis API (*Pachycondyla-apicalis*) afin de l'utiliser pour assurer la couverture dans les RCSF en minimisant le nombre des nœuds de capteur utilisé.

3.2 Définition d'heuristique :

Une heuristique d'optimisation est une méthode approchée se voulant simple, rapide et adaptée un problème d'optimisation donné. Sa capacité à optimiser un problème avec un minimum d'informations est contre balancé par le fait qu'elle n'offre aucune garantie quant à l'optimalité de la meilleure solution trouvée [42].

3.3 Définition de méta-heuristique :

Parmi les heuristiques, certaines sont adaptables a un grand nombre de problèmes différents sans changements majeurs dans l'algorithme, on parle alors de méta-heuristiques. La plupart des heuristiques et des méta-heuristiques utilisent des processus aléatoires comme moyens de récolter de l'information et de faire face à des problèmes combinatoire. En plus de cette base stochastique, les méta-heuristiques sont généralement itératives, c'est-à-dire qu'un même schéma de recherche est appliqué plusieurs fois au cours de l'optimisation et directes [42]. Dans la littérature on trouve de deux types de méthodes (méta-heuristiques) : les méthodes trajectoires et les méthodes évolutionnistes. Ces dernières sont détaillées par la suite.

3.4 Les méthodes trajectoires :

L'espace de recherche associé à un problème d'optimisation combinatoire est souvent non énumérable en un temps raisonnable. On essaie donc de relier certaines solutions entre elles. Ainsi, à partir d'une solution, on peut en trouver une autre et ainsi de suite. Il est nécessaire de définir une relation de voisinage qui est une application qui associe à toute solution de l'espace de recherche un voisinage (i.e. un ensemble de solutions ne la contenant pas elle-même) appelée voisins. Les recherches locales sont des méthodes fondées sur une relation de voisinage et sur une procédure exploitant ce voisinage. Les recherches locales se différencient par la procédure d'exploitation du voisinage, le voisinage pouvant être considéré comme un paramètre de celle-ci. Dans ces méthodes on peut trouver trois méthodes différentes : La méthode de descente [43], la méthode de recuit simulé (Simulated Annealing) [44], la méthode (la recherche) taboue (Tabu Search) [45].

3.5 Les méthodes évolutionnistes :

Les algorithmes évolutionnistes ou algorithmes évolutionnaires (evolutionary algorithms en anglais), sont une famille d'algorithmes dont le principe s'inspire de la théorie de l'évolution pour résoudre des problèmes divers. Ce sont donc des méthodes de calcul bio inspirées. L'idée est de faire évoluer un ensemble de solutions à un problème donné, dans l'optique de trouver des résultats proches. Ce sont des algorithmes dits stochastiques, car ils utilisent itérativement des processus aléatoires.

La grande majorité de ces méthodes sont utilisées pour résoudre des problèmes d'optimisation combinatoire, bien que le cadre général ne soit pas nécessairement dédié aux algorithmes d'optimisation au sens strict. Comme méthodes évolutionnistes on cite : les algorithmes génétiques et les colonies de fourmis. Ces derniers sont présentés dans la suite de cette section.

3.5.1 Les algorithmes de colonies de fourmis :

Les algorithmes de colonie de fourmis sont inspirés à partir des insectes sociaux où le comportement de la vraie fourmi est exploité, résolvent naturellement des problèmes complexes. Les fourmis communiquent indirectement les unes avec les autres en déposant des produits chimiques appelés phéromones sur le sol. Ce genre de la communication indirect s'appelle la stigmergie. En effet, sur la base de l'utilisation de cette substance chimiques, l'ensemble des fourmis et après l'étape de recherche effectuée, ils ont tous tendance à prendre le plus court chemin entre le nid et l'obstacle. Plus le taux de phéromone est élevé dans un lieu donné, plus la fourmi a plus de chances d'être attirée par cette zone. Les fourmis qui ont retourné le plus rapide au nid en passant par la source de nourriture ont choisi le plus court

chemin qui implique que la quantité de phéromone déposée sur ce chemin est plus grande. Par conséquent ce plus court chemin a une grande probabilité d'être prise par les fourmis que les autres chemins [46].

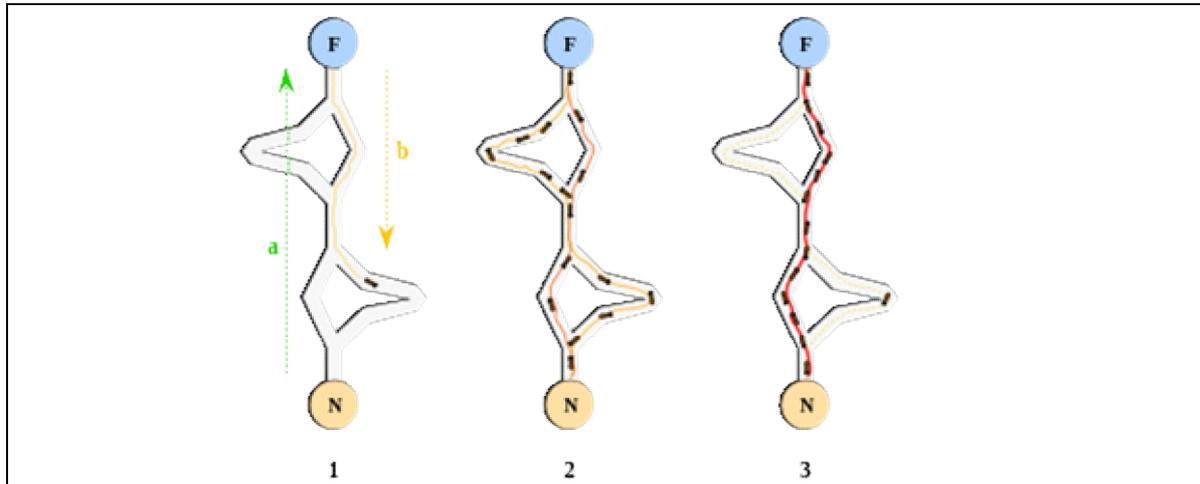


Figure 3.1 :Algorithme de colonies de fourmis.

Le premier algorithme de colonie de fourmis est introduit pour résoudre le problème de voyageur de commerce afin de trouver le plus court chemin en déplaçant entre plusieurs villes.

3.5.1.1 Algorithme (Système de fourmi : Ant System (AS)) :

Il convient de noter que dans ce problème de voyageur de commerce, il est nécessaire de trouver le plus court chemin reliant n villes, où chaque ville est visitée qu'une seule fois. Ce problème est l'un des premiers qui a déclenché l'utilisation des fourmis afin de le résoudre. En ce qui concerne l'optimisation de la distance à parcourir. Un certain nombre d'études ont montré que les vraies fourmis confrontées à ce genre de problèmes peuvent apporter une réponse. La liste des villes à visiter est connue, et l'algorithme est basé sur des chemins contenant de la substance de phéromones. Cette application est fournie des résultats très satisfaisants [47].

1. Les étapes de l'algorithme « Ant System »: (AS) pour résoudre le problème du voyageur de commerce :

- 1) Chaque fourmi choisit un trajet, et trace une piste de phéromone.
- 2) l'ensemble des fourmis parcourt un certain nombre de trajets où chaque fourmi dépose une quantité de phéromone proportionnelle à la qualité du parcours.
- 3) chaque arête du meilleur chemin est renforcée plus que les autres.
- 4) l'évaporation de phéromone déposée fait disparaître les solutions mauvaises.

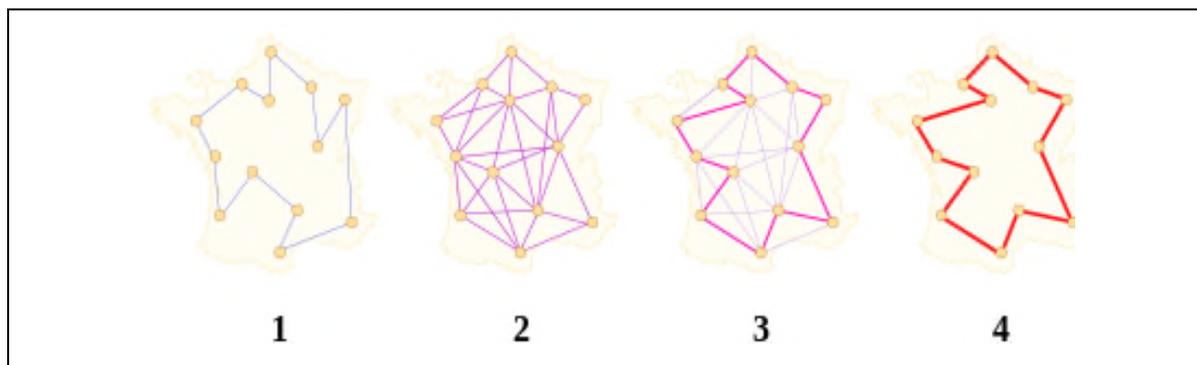


Figure 3.2 : Les étapes utilisées par l'algorithme AS pour déterminer le chemin optimal.

3.5.1.2 Système de colonie de fourmis (ACS) : représente l'amélioration de l'algorithme Ant system (AS).

a) Première amélioration :

Dans les grands parcours, on peut reprocher à AS de ne pas se focaliser suffisamment sur les bonnes solutions précédemment obtenues afin d'influer sur les décisions des fourmis. Pour arriver à cela ACS utilise une variable constante ψ_0 paramétrable générée à partir de $[0, 1]$. Ils ont attribué une valeur aléatoire uniformément produite de l'intervalle $[0, 1]$ au hasard à la variable ψ . Quand $\psi \leq \psi_0$, le choix est basé uniquement sur les villes qui sont déjà visitées (effectivement). Par contre, si $\psi > \psi_0$, on maintient le même choix que le AS. Avec l'ajustement de ψ_0 , les fourmis sont orientées de trouver les meilleures solutions ou les guidés à découvrir d'autres chemins.

b) Seconde amélioration :

ACS exploite une politique de mise à jour de la quantité de phéromones déposée afin d'explorer la meilleure solution. Pendant que la mise à jour localement est effectuée (la mise à jour pas à pas est effectuée sur un arc lorsque une fourmi l'emprunte), l'ACS réduit la quantité de phéromones. Contrairement à AS, ACS augmente lors de la mise à jour globale (après chaque cycle) uniquement le chemin auquel il appartient dans le meilleur trajet (une de la longueur minimale) [48].

3.5.1.3 Max-Min Ant System (MMAS):

a) Différences avec AS :

1. Uniquement le parcours le plus court est mis à jour en phéromones.
2. Les valeurs des phéromones sur chaque arête sont bornées par T_{\min} (la quantité de phéromone minimale) et T_{\max} (la quantité de phéromone maximale).

3. Les valeurs des phéromones sur chaque arête sont initialisées à la valeur maximum T_{\max} .
4. La quantité de phéromones que l'on fait évaporer est proportionnelle à sa valeur au moment de la modification, plus les pistes sont fortes plus ses phéromones seront diminués.

b) Intérêt :

On empêche ainsi la monopolisation de certaines arêtes qui ont été tellement imprégnés au début du processus de recherche qu'ils sont systématiquement parcourus par les fourmis. On permet grâce à cette façon de gérer l'évaporation de tirer vers le bas, les arêtes chargées fortement en phéromones afin de vérifier si leur importance est pertinente. De ce fait si ce n'est pas le cas, les pistes plus faiblement chargées pourront leur prendre le pas [49].

3.5.1.4 L'algorithme Pachycondyla-apicalis (API) :

L'algorithme (API) est proposé par N. Monomarché en 2000 [50], qui simule le comportement d'un type de fourmi trouvé en Amérique du Sud appelée Pachycondyla-apicalis. Dans l'algorithme API, la collaboration entre les individus se reflète dans la division de l'espace de recherche contrairement à l'utilisation de la technique de stigmergie exploitée par la plupart des algorithmes de colonie de fourmis proposés, où la solution globale est celle qui attire vers elle le plus d'individus. Les espèces sont caractérisés par une colonie peu peuplée (quelques dizaines d'individus), un habitat instable se développe souvent et manque de communication directement entre les individus. Ces propriétés ont été mentionnées dans la modélisation d'algorithme d'API.

1) Principe de l'algorithme :

La colonie envoie n fourmis a_1, \dots, a_n (dites fourrageuses) autour du nid. Chaque fourmi crée puis garde en mémoire p sites de chasse. En parallèle des autres fourmis, la fourmi choisit aléatoirement un de ses sites de chasse noté S , et commence une exploration locale. Si l'exploration est fructueuse, elle remplace S par son nouveau site de chasse S_0 . Sinon, elle compte un échec supplémentaire pour le site S . Si le nombre d'échecs de S dépasse un seuil appelé patience locale P_{locale} , il oublie et le remplace par un nouveau site aléatoire. La patience permet de creuser une piste pendant quelques tours plutôt que de l'abandonner tout de suite. La colonie rappelle régulièrement ses fourrageuses et regarde leurs résultats. Si un site de chasse d'une fourmi est meilleur que ceux des autres fourmis et que la position actuelle du nid, tout le nid se déplace et la mémoire de toutes les fourmis repart de

zéro. La procédure recommence alors : les fourmis sont de nouveau disséminées autour du nid.

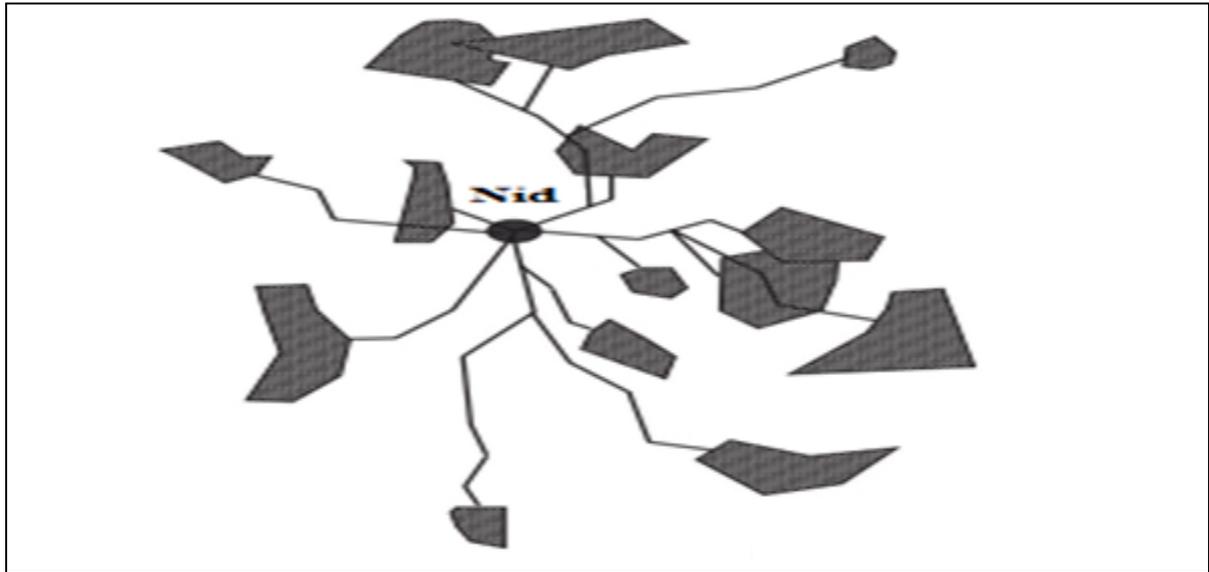


Figure 3.3 : Exploration des sites de nourriture par les fourmis Pachycondyla-apicalis.

1.1) Les Notations utilisées par l'algorithme API:

1. Sr : un espace de recherche quelconque.
2. $N \in Sr$: la position du nid, initialisée avec l'opérateur aléatoire O_{rand} .
3. O_{explo} retourne un point S_0 dans le voisinage d'un point $S \in Sr$.
4. $f : Sr \rightarrow R$: la fonction objectif à minimiser.
5. $n_s(a_i)$: le nombre de sites de chasse de la fourmi a_i .
6. A_{site} et A_{locale} : les amplitudes maximales pour créer les sites de chasse.
7. e_j : le nombre d'échecs associé au site de chasse S_j pour une fourmi donnée.
8. P_{locale} : le seuil d'échecs admissible avant qu'une fourmi n'abandonne pas un site.

1.2) Les différentes étapes de l'algorithme : Les principales étapes de la simulation de la colonie de fourmis **Pachycondyla-apicalis** sont données par l'algorithme suivant :

Algorithme 1 : Simulation de la colonie : API ()

```

1 Choisir l'emplacement initial du nid :  $\mathbf{N} := \mathbf{O}_{\text{rand}}$  ;
2  $\mathbf{T} := 0$  ; // Nombre d'exploration des fourmis
3 tant que la condition d'arrêt n'est pas vérifiée faire
4   Pour chaque fourmi  $\mathbf{a}_i$  faire
5      $\text{API\_FOURRAGEMENT}(\mathbf{a}_i)$  ;
6     Si le nid doit être déplacé alors
7        $\mathbf{N} := \mathbf{S}^+$  ; // Meilleure solution trouvée par une fourmi
8       Vider la mémoire de toutes les fourmis ;
9      $\mathbf{T} := \mathbf{T} + 1$ 
10 retourné  $(\mathbf{S}^+, \mathbf{f}(\mathbf{S}^+))$ 

```

Au départ, et à chaque changement de la position du nid, chaque fourmi \mathbf{a}_i quitte le nid pour se constituer une liste de sites de chasse qu'elle mémorise. Un site de chasse est un point de \mathbf{S}_r construit par l'opérateur $\mathbf{O}_{\text{explo}}$ avec une amplitude $\mathbf{A}_{\text{site}}(\mathbf{a}_i)$ dans le voisinage de \mathbf{N} . La fourmi \mathbf{a}_i va ensuite procéder à une exploration locale autour d'un de ses sites de chasse. Initialement, quand l'intérêt des sites est inconnu, la fourmi choisit un site au hasard parmi les \mathbf{p} dont elle dispose. L'exploration locale consiste à construire un point de \mathbf{S}_0 dans le voisinage de \mathbf{S} grâce à l'opérateur $\mathbf{O}_{\text{explo}}$ avec une amplitude $\mathbf{A}_{\text{locale}}(\mathbf{a}_i)$. La fourmi \mathbf{a}_i capture une proie si cette exploration locale a permis de trouver une meilleure valeur de \mathbf{f} , ce qui revient à avoir $\mathbf{f}(\mathbf{S}_0) < \mathbf{f}(\mathbf{S})$. Une amélioration de \mathbf{f} modélise donc la capture d'une proie. A chaque fois qu'une fourmi parvient à améliorer $\mathbf{f}(\mathbf{S})$ elle mémorise \mathbf{S}_0 à la place de \mathbf{S} et sa prochaine exploration locale aura lieu dans le voisinage de \mathbf{S}_0 . Si l'exploration locale est infructueuse, pour la prochaine exploration, la fourmi choisira un site au hasard parmi les \mathbf{p} sites qu'elle a en mémoire. Quand un site a été exploré successivement plus de $\mathbf{P}_{\text{locale}}$ fois sans avoir rapporté de proie, il est définitivement oublié et sera remplacé par un nouveau site à la prochaine itération (c'est-à-dire la prochaine sortie du nid). Le paramètre $\mathbf{P}_{\text{locale}}$ représente une patience locale. À chaque sortie, la colonie envoie les \mathbf{n} fourrageuses en exploration. Les modalités de l'exploration locale sont présentées dans l'algorithme suivant :

Algorithme 2 : Simulation de la fourmi a_i : API_FOURRAGEMENT (a_i)

```

1   Si  $n_S(a_i) < p$  alors
2   // La mémoire de la fourmi n'est pas pleine
3    $n_S(a_i) := n_S(a_i) + 1$  ;
4   Création d'un site de chasse  $S$  autour du nid :  $n_S(a_i) := O_{\text{explo}}(N, A_{\text{site}})$  ;
5   Initialisation du compteur d'échecs pour le site :  $e_{n_S}(a_i) := 0$  ;
6   Sinon
7   Soit  $S_j$  le dernier site exploré par la fourmi ;
8-  Si  $e_j > 0$  alors
9   // La dernière exploration a été infructueuse
10  // Choisir aléatoirement un site de chasse  $S_j$  ( $j \in \{1, \dots, p\}$ )
11  Exploration locale autour de  $S_j$  :  $S_0 := O_{\text{explo}}(S_j, A_{\text{locale}})$  ;
12  Si  $f(S_0) < f(S)$  alors
13  //  $S_j := S_0$  ;
14  //  $e_j := 0$  ;
15  Sinon
16  //  $e_j := e_j + 1$  ;
17  Si  $e_j > P_{\text{locale}}$  alors
18  // Effacer le site  $S_j$  de la mémoire de fourmi ;
19  //  $n_S(a_i) := n_S(a_i) - 1$  ;

```

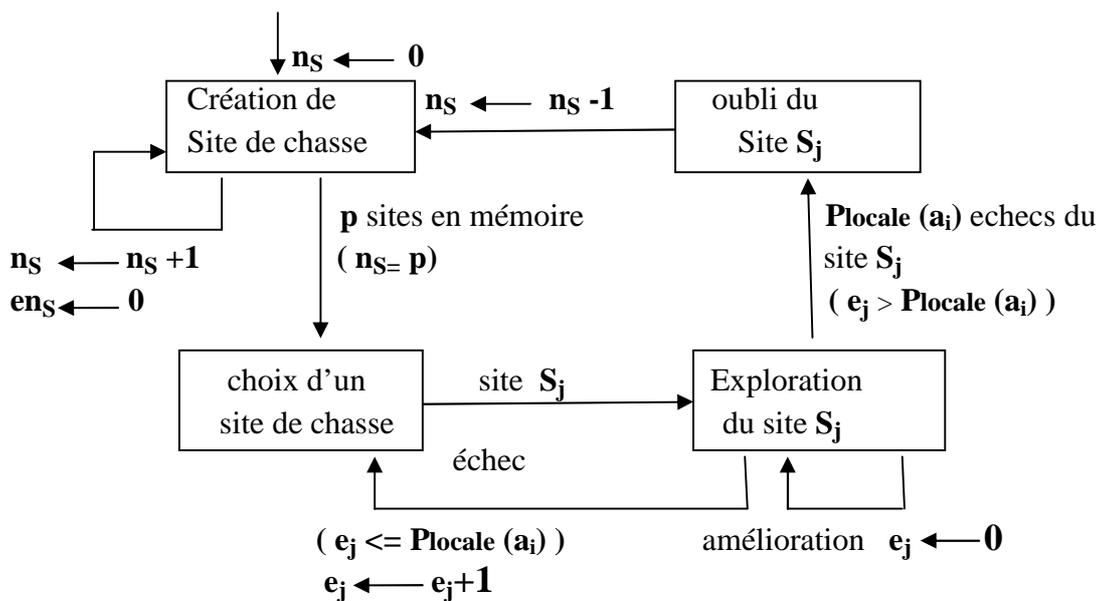


Figure 3.4 : Le comportement individuel de fourragement d'une fourmi. (n_s est le nombre de sites de chasse dans la mémoire de la fourmi et e_j le nombre d'échecs successifs rencontrés sur le site s_j).

L'automate de la figure 3.5 résume le comportement individuel d'une fourrageuse. n_s représente le nombre de sites que la fourmi mémorise à un instant donné e_j donne le nombre d'échecs successifs du site S_j mémorisé.

2) Application de l'algorithme API :

Dans cette section, nous exposons quelques applications de l'algorithme **API** à des différents problèmes d'optimisation dont l'ensemble de recherche peut contenir différentes formes. Cela peut être réellement appliqué à partir du moment où les opérateurs **Orand** et **Oexplo** sont définis sur l'espace de recherche [50].

2.1. Opérateurs spécifiques à l'optimisation numérique :

Dans cette section nous détaillons les opérateurs **Orand** et **Oexplo** définis dans la section précédente d'une façon générale afin de montrer comment ils ont été adaptés à différents problèmes. Les fonctions étudiées sont définies sur un produit d'intervalle de \mathbf{R} ; chacun délimité par deux bornes \mathbf{b}_i et \mathbf{B}_i . L'opérateur **Orand** doit donc générer de façon aléatoire un point de l'espace $\mathbf{S} = [\mathbf{b}_1, \mathbf{B}_1] \times \dots \times [\mathbf{b}_l, \mathbf{B}_l]$ où l représente la dimension de la fonction à minimiser. Le point $\mathbf{S} = (s_i) \quad i=1..l$ tel que \mathbf{S} est généré de façon uniforme :

$$s_i = \mathbf{b}_i + U [0,1] \times (\mathbf{B}_i - \mathbf{b}_i)$$

Où $U [0,1]$ est un nombre aléatoire uniformément tiré dans $[0,1]$.

L'opérateur d'exploration **Oexplo** tient compte d'une amplitude \mathbf{A} :

$$s'_i = s_i + \mathbf{A} \times U [-0.5, 0.5] \times (\mathbf{B}_i - \mathbf{b}_i) \quad i \in \{1, \dots, l\}$$

Où $U [-0.5, 0.5]$ est un nombre aléatoire uniformément tiré dans $[-0.5, 0.5]$. Evidemment, il faut s'assurer que $s'_i \in [\mathbf{b}_i, \mathbf{B}_i]$. Cet opérateur a lui aussi un comportement uniforme dans l'intervalle $[s_i - 0.5 \times \mathbf{A} (\mathbf{B}_i - \mathbf{b}_i), s_i + 0.5 \times \mathbf{A} (\mathbf{B}_i - \mathbf{b}_i)]$.

2.2. Optimisation de fonctions numériques :

Pour ce problème, l'espace de recherche \mathbf{S} est un produit d'intervalle de \mathbf{R} . Tous les tests sont basés sur un ensemble de fonctions continues définies sur \mathbf{R}^n dans \mathbf{R}^+ . Pour chacune de ces fonctions le problème d'optimisation consiste à minimiser la fonction.

Le minimum est connu par avance et vaut $\mathbf{0}$ pour l'ensemble des fonctions. Pour les deux opérateurs **Orand** et **Oexplo**, sont définis dans le point précédent (**Opérateurs spécifiques à l'optimisation numérique**).

Optimisation combinatoire (problème du voyageur de commerce) :

Le problème du voyageur de commerce (**PVC**, Traveler Salesman Problem: **TSP**) est un problème très classique en optimisation combinatoire. Cependant il est intéressant de voir comment **API** peut s'appliquer simplement à ce problème. De nombreux algorithmes inspirés

des fourmis sont nés de l'analogie de ce problème et la recherche de nourriture par les fourmis réelles.

API pour le PVC :

L'adaptation la plus immédiate d'API à la résolution de problèmes combinatoires est de procéder par la modification successive d'un ensemble de solutions, contrairement à une approche constructive des solutions. L'espace de recherche \mathbf{S} l'ensemble de permutation de n villes, la position d'une fourmi dans l'espace \mathbf{S} est équivalente à un chemin hamiltonien dans le graphe des villes (c-à-d: la position d'une fourmi dans \mathbf{S} est une séquence de ville).

Orand génère une séquence de ville au hasard, le déplacement d'une fourmi correspond alors à une modification de ce chemin. Un certain nombre d'heuristiques sont envisageables pour accomplir cette modification. Dans [51], ils ont expérimenté quelques-unes avec la contrainte qu'elles soient paramétrables par une amplitude. Cela signifie qu'une valeur élevée de l'amplitude permet de modifier assez fortement une solution alors qu'une valeur faible ne doit pas trop perturber un chemin. La plupart des heuristiques efficaces utilisent des heuristiques proches du problème permettant de modifier un chemin pour en trouver un plus court. L'algorithme **ACS**, par exemple (voir la section 3.5.1.2), construit des solutions en utilisant la coopération des fourmis puis chaque chemin est amélioré (si possible) par d'autres heuristiques. Voici les techniques de modification d'un chemin, correspondant à l'opérateur **Oexplo** expérimentées :

- **La permutation simple** : deux villes sont échangées au hasard dans la séquence. L'amplitude correspond dans ce cas au nombre de permutations effectuées (une amplitude de 0.1 pour un problème à 100 villes occasionnera $0.1 \times 100 = 10$ permutations)
- **La permutation circulaire** : plusieurs villes sont permutées dans la séquence, le nombre de villes permutées représente l'amplitude du déplacement d'une fourmi.
- **L'insertion** : cette méthode consiste à choisir une ville au hasard et à l'insérer entre deux autres villes de la séquence. L'amplitude correspond alors au nombre d'insertions effectuées.
- **Permutation probabiliste** : si la permutation de deux villes choisies au hasard dans la séquence augmente la longueur du chemin, la probabilité que cette permutation soit effectuée est inversement proportionnelle à l'augmentation occasionnée. Si la permutation diminue la longueur du chemin, elle est systématiquement effectuée. Ces méthodes de perturbation d'une séquence ont le mérite d'être simples et peu coûteuses au niveau du temps de calcul. On peut cependant utiliser des méthodes améliorant localement une solution comme technique de déplacement d'une fourmi, par exemple on peut appliquer une méthode gloutonne sur une

portion du chemin. On peut également envisager de différencier la création d'un site de chasse de l'exploration d'un site et associer à ces deux tâches des techniques différentes de modification de la séquence.

Plus de ces applications citées dans cette section, l'algorithme **API** est appliqué pour l'apprentissage des autres problèmes tels que : l'apprentissage de Chaîne de Markov Cachées [52] et l'apprentissage de Réseaux de Neurones Artificielles [53].

3.6. Conclusion :

Dans ce chapitre, nous avons présenté premièrement deux types des méta-heuristiques utilisées dans l'optimisation combinatoire: les méthodes trajectoires et les méthodes évolutionnistes (les algorithmes de colonie de fourmis) puis nous avons exposé un état de l'art sur les différents algorithmes de colonie de fourmis. En fin; nous avons décrit en détail l'algorithme API (API-Apicalis).

Chapitre 04

Implémentation

Chapitre 04

Implémentation

4.1 Introduction :

Ce chapitre est consacré à la partie pratique de notre travail où nous avons débuté par donner l'objectif de notre travail, puis nous avons décrit le langage de simulation utilisé ainsi que la description et la formulation de problème utilisée pour résoudre de la 1-couverture utilisée dans les RCSF afin d'améliorer sa durée de vie. Enfin, nous avons décrit notre simulateur réalisé, notre implémentation ainsi que les résultats trouvés.

L'objectif de notre travail est d'adapter l'algorithme de colonie de fourmis API (API-Apicalis) au problème de couverture simple (1-couverture) des cibles dans le RCSF afin de sélectionner un nombre minimum de nœuds de capteurs actifs parmi un grand nombre de nœuds de capteurs déployés de manière aléatoire. Les nœuds de capteurs sélectionnés doivent couvrir tout le nombre prédéfini de points cibles, de manière périodique ou continue. Simultanément, ces nœuds de capteurs actifs doivent maintenir la connectivité entre eux et la station de base (SB) de sorte que les précieuses données détectées puissent être transmises à la SB selon le plus court chemin.

4.2 L'environnement de simulation utilisé :

Pour réaliser notre objectif de travail on a utilisé l'environnement de simulation MATLAB. Ce dernier est une application scientifique interactive orientée au calcul vectoriel et matriciel avec une puissante librairie de visualisation. Elle peut résoudre des problèmes de calcul très complexes d'une façon simple et rapide comparée aux langages de programmation traditionnels du type C ou FORTRAN. Le nom MATLAB est dérivé de l'anglais MATrix LABoratory. MATLAB permet le travail interactif soit en mode commande, soit en mode programmation; tout en ayant toujours la possibilité de faire des visualisations graphiques.

MATLAB possède les particularités suivantes par rapport à ces langages :

- la programmation facile.
- la continuité parmi les valeurs entières, réelles et complexes.
- la gamme étendue des nombres et leurs précisions.
- la bibliothèque mathématique très compréhensive.
- l'outil graphique qui inclut les fonctions d'interface graphique et les utilitaires.

- la possibilité de liaison avec les autres langages classiques de programmations (C ou Fortran).

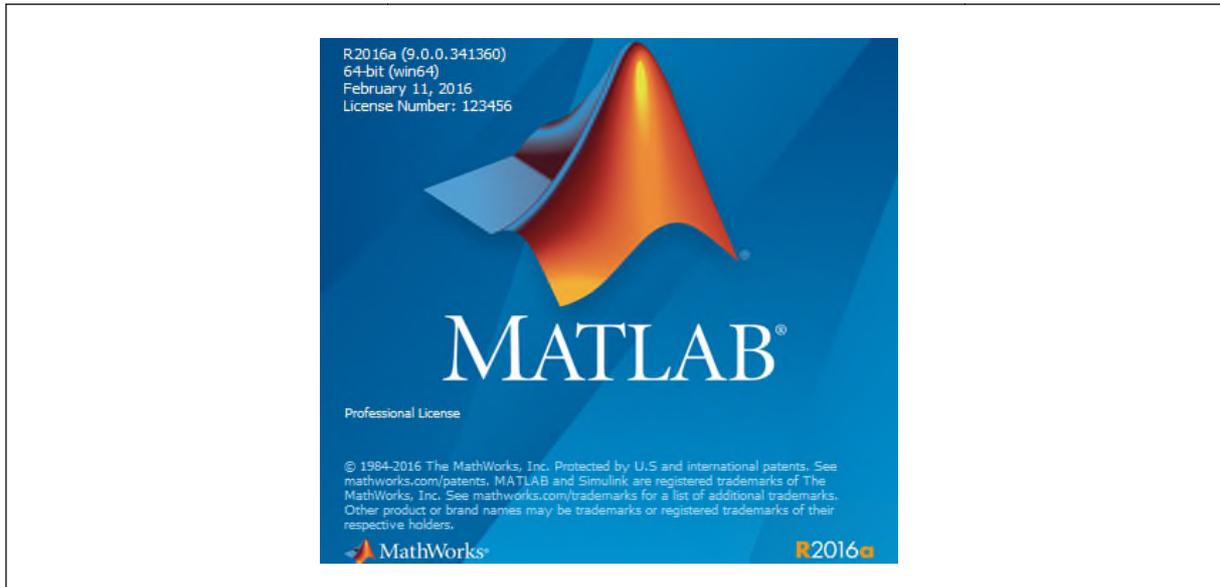


Figure 4.1 : Une fenêtre du MATLAB.

4.3 La modélisation du problème utilisée:

Dans notre travail et afin d'adapter l'algorithme de colonie de fourmis API nous avons utilisé la modélisation prise de [54]. On considère un RCSF (figure 4.2) avec un ensemble V de N nœuds de capteur, $V = \{v_1, v_2, v_3, \dots, v_N\}$ sont déployés de manière aléatoire à l'intérieur de la région d'application pour surveiller de manière continue ou périodique un ensemble de K nombre de cibles C , avec $C = \{c_1, c_2, c_3, \dots, c_K\}$. Un nœud de capteur peut détecter les cibles uniquement si elles se trouvent dans la plage de détection (rayon) R_d des nœuds de capteur. Les nœuds de capteur collectent les données ciblées et les envoient directement à la station de base directe ou par l'intermédiaire d'autres nœuds de capteur. Les nœuds de capteur peuvent communiquer les uns avec les autres uniquement s'ils se trouvent dans la plage de communication (rayon) R_c les uns des autres.

Afin d'améliorer la durée de vie du RCSF, seul un ensemble sélectionné de nœuds de capteur de V est activé pour surveiller tout le nombre K de cibles où chaque nœud surveille 1 cible. De toute évidence, l'ensemble de nœuds de capteurs sélectionnés doit maintenir la connectivité entre eux ainsi que la SB pour le transfert de données. Dans ce cas, au lieu d'activer tous les nœuds de capteur, l'ensemble de nœuds de capteur sélectionné n'est activé que jusqu'à ce que le réseau reste opérationnel. Un nouvel ensemble de nœuds de capteurs est activé à partir du nombre restant de nœuds de capteurs si l'épuisement complet de l'énergie du premiers pour un ou plusieurs. Le processus se poursuit jusqu'à ce qu'un ensemble ne puisse

plus être formé afin de fournir une couverture complète aux cibles et une connectivité entre les nœuds de capteur et la station de base.

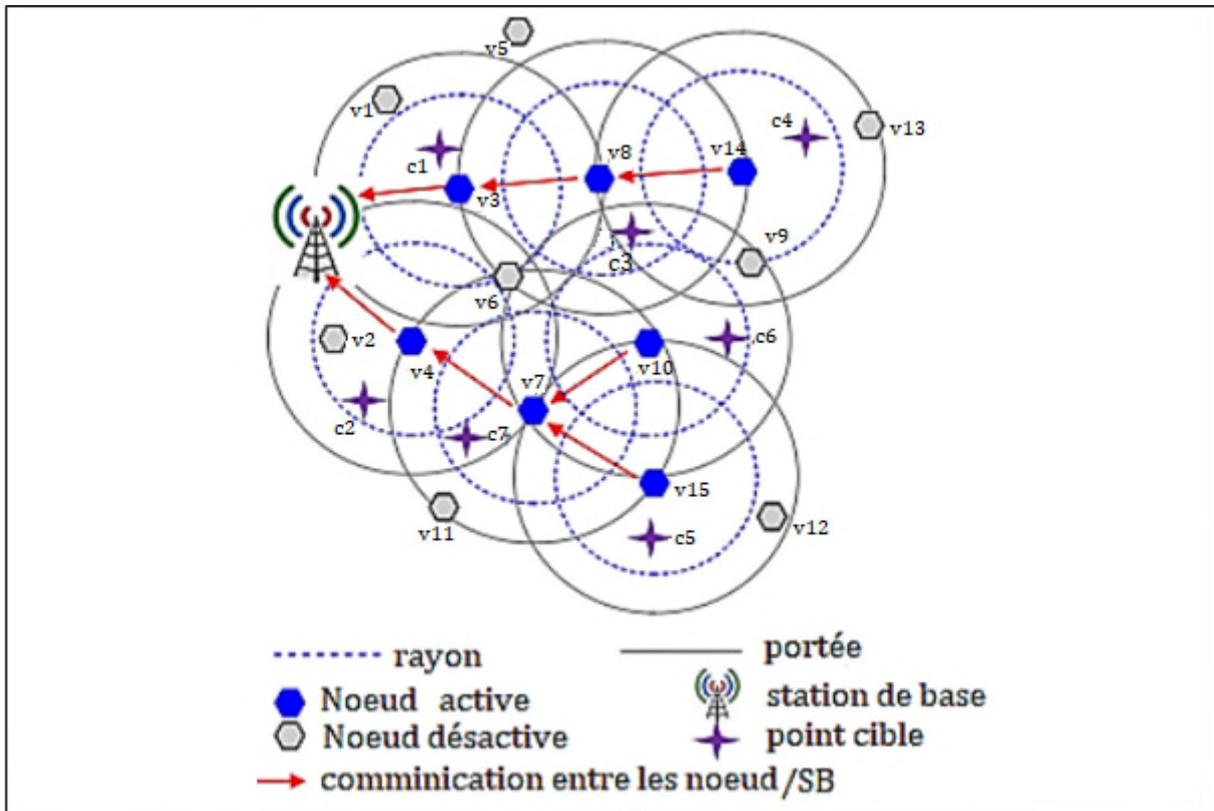


Figure 4.2 : Un exemple de réseau entièrement couvert et connecté.

4.3.1 Le modèle mathématique utilisé :

Le modèle utilisé est pris de [54].

- **Les hypothèses du réseau :**

- La station de base située au hasard dans le champ de détection. Les nœuds de capteurs et la SB sont tous stationnaires après le déploiement.
- Tous les nœuds de capteurs ont la même énergie initiale, tandis qu'une quantité illimitée d'énergie est définie pour la SB.
- Les liens sont symétriques, c'est-à-dire que tout nœud de capteur peut calculer la distance approximative à un autre nœud.
- La collecte périodique de données est effectuée pour les RCSF dans lesquels chacun nœud de capteur génère et envoie la même quantité de données par temps unitaire à la SB par transmission multi-saut.
- La valeur du rayon de détection est inférieure ou égale à celle de rayon de communication pour un nœud de capteur spécifique, bien qu'ils ne soient pas égaux dans certains cas.

• **Le modèle mathématique utilisé:**

N : Nombre de nœud.

K : Nombre de cible.

➤ **Les variables de décision :**

$$X_i = \begin{cases} 1, & \text{si le nœud } v_i \text{ est sélectionné.} \\ 0, & \text{sinon.} \end{cases} \quad (1)$$

$$Y_{ij} = \begin{cases} 1, & \text{si la cible } c_i \text{ est dans la plage de détection } (R_d) \text{ du nœud } v_j. \\ 0, & \text{sinon.} \end{cases} \quad (2)$$

$$K_{ij} = \begin{cases} 1, & \text{si } v_j \text{ est dans la plage de communication } (R_c) \text{ de } v_i, \\ & \text{et en direction vers SB à partir de } v_i, i \neq j. \\ 0, & \text{sinon.} \end{cases} \quad (3)$$

$$Z_i = \begin{cases} 1, & \text{si } v_i \text{ est dans la plage de communication } (R_c) \text{ de SB.} \\ 0, & \text{sinon.} \end{cases} \quad (4)$$

➤ **La fonction objectif :** $Min F = \sum_{i=1}^N X_i.$ (5)

➤ **Les contraintes :**

$$\sum_{j=1}^N Y_{ij} \times X_j = 1, \forall i, 1 \leq i \leq K, Y_{ij} \in \{0, 1\}, X_i \in \{0, 1\}. \quad (6)$$

$$\left(\sum_{j=1}^K K_{ij} \times X_j + Z_i \right) \times X_i \geq 1, \forall i, 1 \leq i \leq N, K_{ij} \in \{0, 1\}, Z_i \in \{0, 1\}. \quad (7)$$

La contrainte (6) implique que chaque cible c_i , $\forall i, 1 \leq i \leq K$, est couverte par un et un seul nœud de capteur. La contrainte (7) garantit que tous les nœuds de capteur sélectionnés ont au moins un nœud de capteur en tant que nœud de transmission de données ou que le nœud de capteur se trouve dans la plage de transmission directe de la station de base. Les nœuds capteurs v_i sont dans la plage de transmission de la SB est donnée par la variable Z_i .

4.4 L'implémentation

Dans ce point concernant l'implémentation, on va décrire les différents paramètres utilisés par l'algorithme d'API ainsi que les différents paramètres utilisés pour le RCSF.

4.4.1 Les paramètres de l'algorithme d'API utilisés :

Les différents paramètres d'API appliqués pour atteindre notre objectif de travail sont les suivants :

- ✓ Le nombre de fourmis (n) : $n \in \{1, 2, 3, 4, 5\}$.
- ✓ L'amplitude locale (Alocale) : $A_{locale} = 0.1$.
- ✓ La patience locale (Plocale) : $P_{locale} = 20$.
- ✓ Le nombre de sites à mémoriser pour chaque fourmi (P) : $P = 2$.
- ✓ La patience globale autour du nid (P_N) : $P_N = 84$.
- ✓ L'espace de recherche (S) : $S =$ l'ensemble des solutions possibles.

✓ **La détermination des deux opérateurs Orand et Oexplo :**

Les opérateurs **Orand** et **Oexplo** développés pour l'optimisation combinatoire pour le problème du voyageur de commerce (PVC) peuvent être utilisés dans notre implémentation, comme décrit dans le chapitre 03.

Orand : permet de générer aléatoirement une solution s à partir de l'ensemble de solutions réalisables S (un chemin du nœud source vers la station de base avec une 1-couverture en respectant la connectivité entre les nœuds de la solution. Cette solution représente la position du nid).

Oexplo : permet de générer les solutions voisines s_i de la solution s déterminée par l'**Orand**, on utilisant la permutation simple. Dans notre travail **Oexplo** basé seulement sur l'amplitude **Alocal** (voir le chapitre 03).

4.4.2 Les paramètres utilisés pour le RCSF pour la simulation :

Paramètres	Valeurs
La surface de déploiements (m)	500 m x500 m, 600 m x600 m, 700 m x700 m, 900 m x900 m, 1000 m x1000 m
Nombre de nœuds	4, 12, 16, 25, 50
Rayon de transmission	150 m
Rayon de détection	120 m
Energie initial d'un nœud	0.5 J
Nœud source (Emetteur)	1, 2, 1, 1, 10

Tableau 4.1 : Les paramètres utilisés pour la simulation pour le RCSF.

4.4.3 Fonctionnement du simulateur réalisé :

Le fonctionnement de base de notre simulateur est tracé par le l'organigramme de la figure suivante :

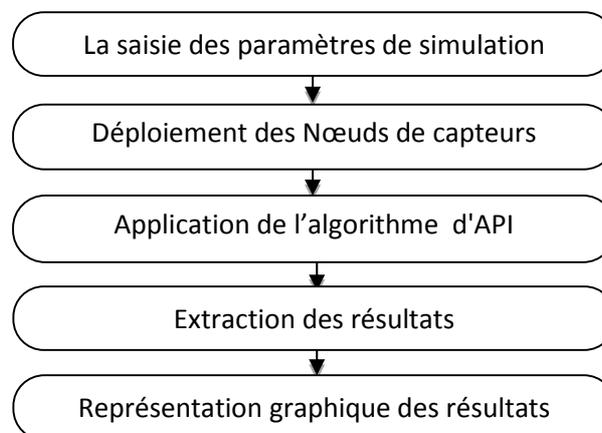


Figure 4.3 : Fonctionnement du simulateur réalisé.

4.4.4 Bloc fonctionnel du simulateur :

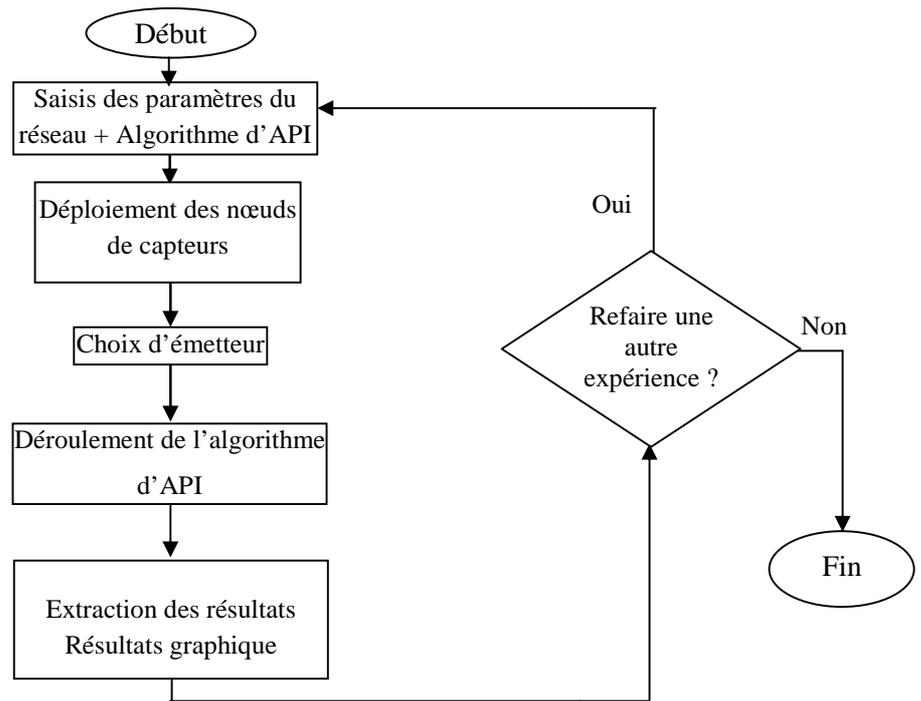


Figure 4.4 : Schéma du bloc fonctionnel du simulateur réalisé.

4.4.5 La description du Simulateur et les résultats de simulation :

Nous présentons dans cette section la description de notre simulateur, on commençant par l'interface principale.

a. L'interface principale : Une fois l'utilisateur lance le simulateur alors il obtient la fenêtre suivante :

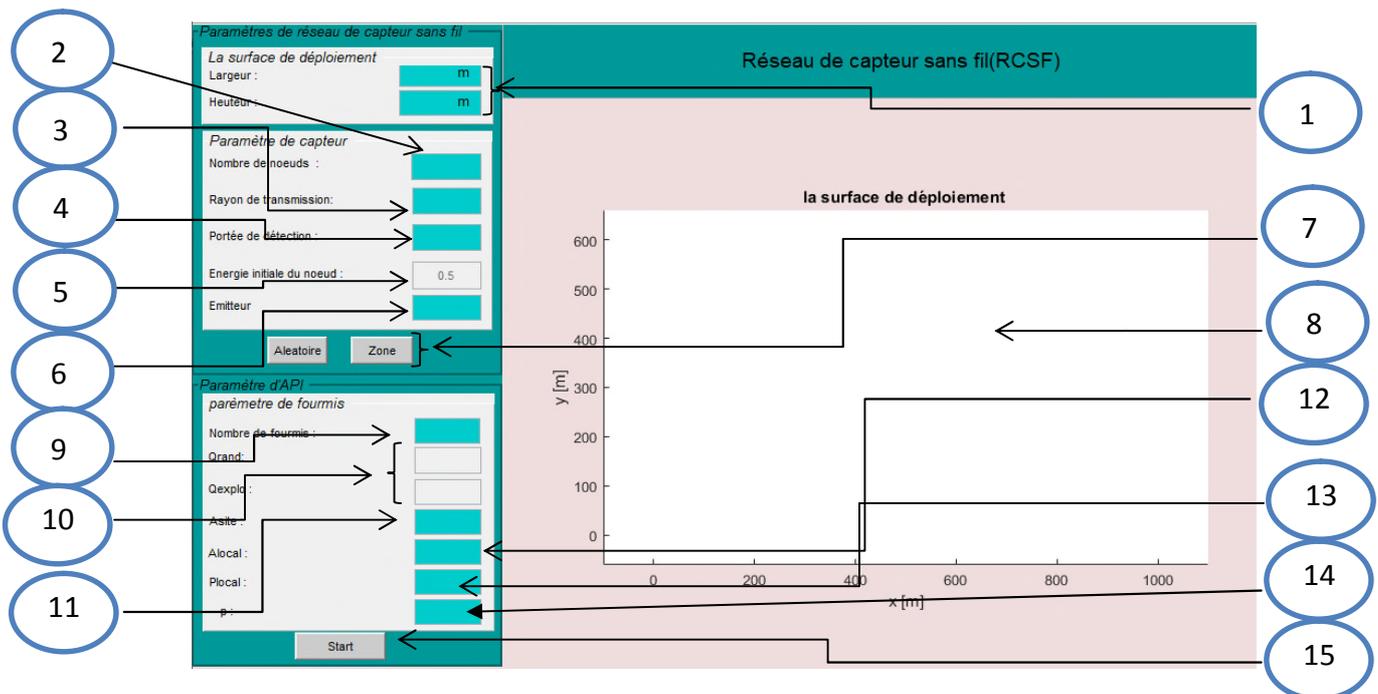


Figure 4.5 :L'interface principale de notre simulateur.

- 1-zone représente la surface de déploiement.
- 2-zone identifier le nombre des nœuds.
- 3-zone pour saisir le rayon de transmission.
- 4-zone pour saisir le rayon de détection.
- 5-zone représente l'énergie initiale de nœud.
- 6-zone pour saisir le nœud émetteur.
- 7-zone contient des boutons de déploiement des nœuds.
- 8-la surface de déploiement.
- 9-zone pour saisir le nombre de fourmis.
- 10-les opérateurs **Oexplo** et **Orand**.
- 11-zone pour saisir **Asite**.
- 12-zone pour saisir **Alocal**.
- 13-zone pour saisir **Plocal**.
- 14-zone pour saisir **p**.
- 15-bouton pour afficher le résultat.

a) Déploiement des nœuds de capteur :

Cette étape consiste à déployer les nœuds de capteur sans fil dans un espace à deux dimensions d'une manière aléatoire qui ne respecte pas l'homogénéité de présence dans l'espace, il est basé sur une fonction aléatoire qui génère à chaque fois un positionnement différent par rapport au positionnement précédent.

Chaque nœud au démarrage possède les données suivantes :

- Une position en deux dimensions.
- Un rayon radio de son transmetteur permettant la détection de ses voisins et rayon de détection permettant la détection des cibles.
- Energie initiale de nœud.

✓ Le déploiement aléatoire des nœuds de capteur :

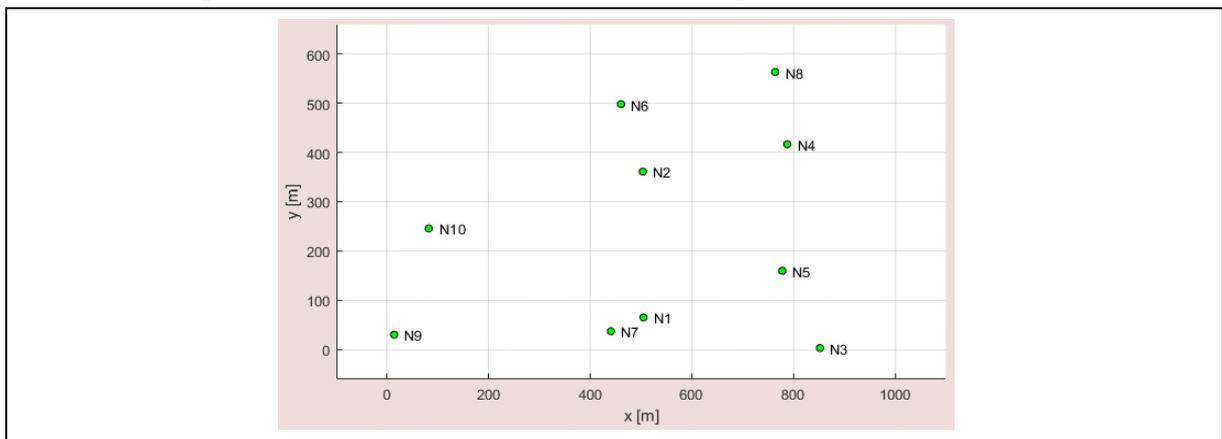


Figure 4.6 : Le déploiement aléatoire des nœuds de capteur.

b) Le déroulement de l'algorithme d'API :

Une fois que les nœuds de capteur du réseau sont déployés on procède à exécuter l'algorithme d'API adapté. Ce dernier est déroulé sur plusieurs exemples présentés par la suite dans cette section.

4.5 Les paramètres d'API :

Paramètres	Descriptions
N	Nombre de fourmis
<i>Orand</i>	Opération générant aléatoirement un point s de S
<i>Oexpl</i>	Opération générant un point s_i dans le voisinage de s
<i>Asite</i>	Amplitude de la création d'un site de chasse
<i>Alocale</i>	Amplitude d'exploration locale
<i>Plocale</i>	Patience locale
P	Le nombre de sites à mémoriser par chaque fourmi

Tableau 4.2 : Les paramètres d'API.

Après la définition des paramètres du RCSF et de l'algorithme d'API adapté pour la couverture simple (1-couverture). On exécute ce dernier sur ce réseau dont les nœuds de capteur sont générés aléatoirement ayant presque les mêmes paramètres tels que : le rayon de transmission, le rayon de communication et l'énergie initiale.

Les grandes lignes de l'algorithme API mise en œuvre sont les suivantes :

- **Début** : déterminer une solution s_0 (nid) à partir de l'ensemble de solutions réalisables S à l'aide de l'opérateur *Orand*.
- **Boucle** : répétez l'exploration du voisinage de la solution s_0 jusqu'à que la condition d'arrêt est vérifiée (nombre d'itérations).
- **Solution** : minimiser le nombre de nœuds de capteur assurant la 1-couverture où la connectivité est assurée entre ces nœuds de capteurs actifs entre la SB de sorte que la transmission des données à la SB est faite selon le plus court chemin.

4.6 Les expérimentations :

Par le biais de plusieurs expérimentations faites et afin d'atteindre notre objectif de ce travail, nous avons variés les différents paramètres utilisés par le RCSF simulé détaillées dans cette section.

- **Première expérimentation :**

Le tableau suivant représente les paramètres utilisés :

Les paramètres	La surface	Nombre de nœuds	Rayon de transmission	Rayon de détection
	600 m x 600 m	4	150 m	120 m

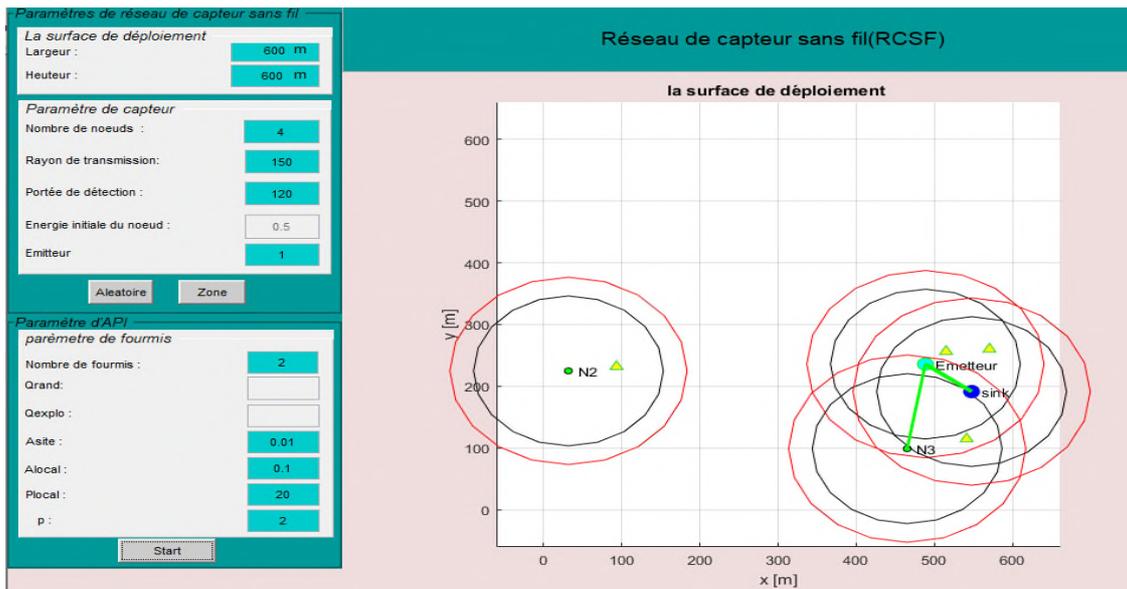


Figure 4.7 : La première expérimentation avec 4 nœuds de capteur et 4 cibles déployés de manière aléatoire.

De la figure 4.7 et après l’exécution de l’algorithme d’API sur le réseau défini et avec le choix du premier nœud (N1) comme un nœud émetteur, on remarque que l’algorithme d’API a déterminé le plus court chemin direct (mono-saut) reliant l’émetteur avec le sink. Le nœud N3 bascule en mode veille, car il se trouve dans le rayon de transmission de l’émetteur.

- **Deuxième expérimentation:** Le tableau suivant représente les paramètres utilisés :

Les paramètres	La surface	Nombre de nœuds	Rayon de transmission	Rayon de détections
	700 m x 700 m	12	150 m	120 m

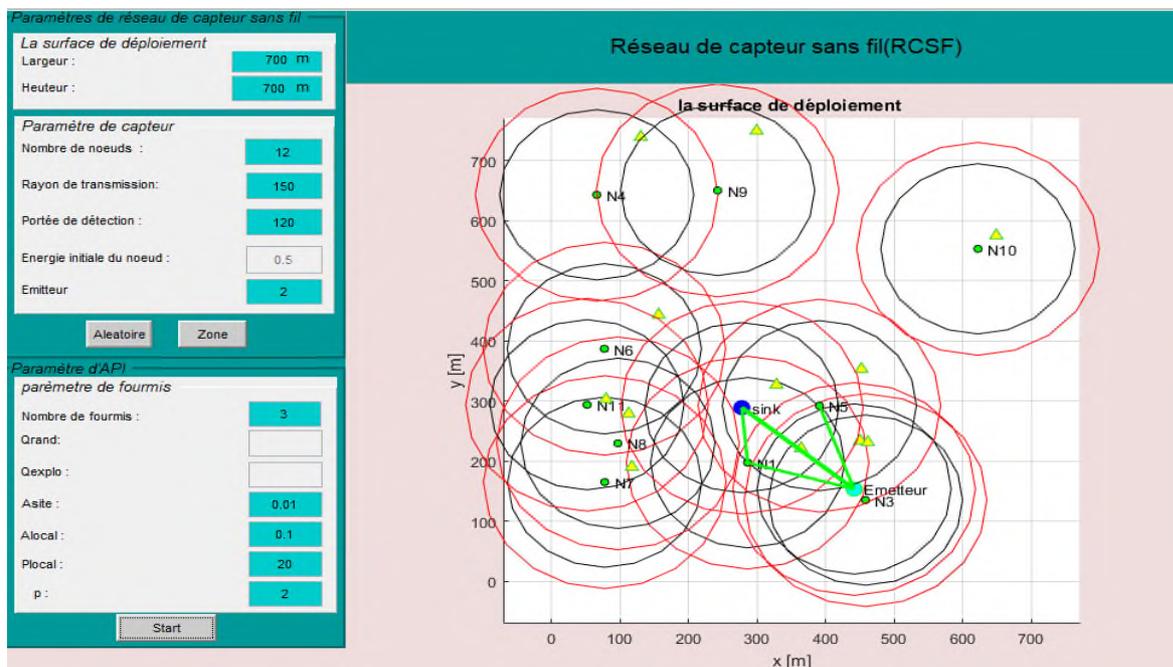


Figure 4.8 : La deuxième expérimentation avec 12 nœuds de capteur et 12 cibles déployés de manière aléatoire.

De la figure 4.8 et après l'exécution de l'algorithme d'API sur le réseau défini et avec le choix du deuxième nœud (N2) comme un nœud émetteur, on remarque que l'algorithme d'API adapté à déterminer le plus court chemin indirect (multi-saut) reliant l'émetteur avec le sink. Le nœud (N6) bascule en mode veille, car il se trouve dans le rayon de transmission de l'émetteur.

• **Troisième expérimentation :**

Le tableau suivant représente les paramètres utilisés :

Les paramètres	La surface	Nombre de nœuds	Rayon de transmission	Rayon de détections
	700 m x 700 m	16	150 m	120 m

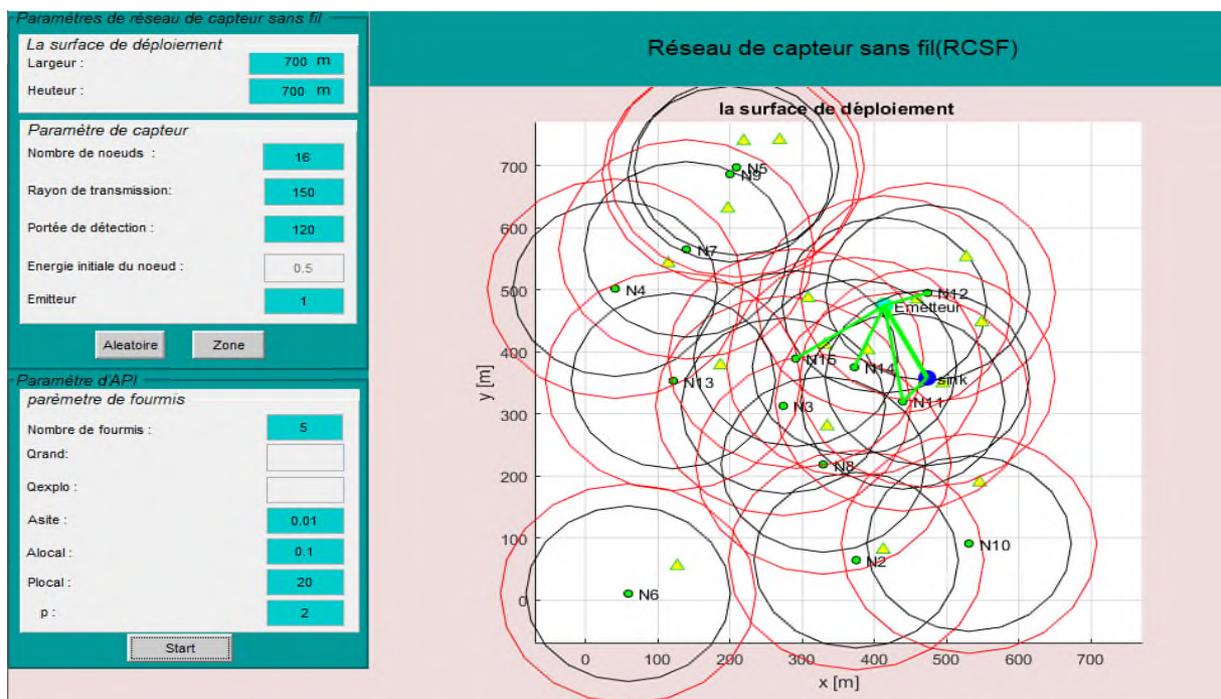


Figure 4.9 : La troisième expérimentation avec 16 nœuds de capteur et 16 cibles déployés de manière aléatoire.

De la figure 4.9 et après l'exécution de l'algorithme d'API sur le réseau défini et avec le choix du premier nœud (N1) comme un nœud émetteur, on remarque que l'algorithme d'API adapté à déterminer le plus court chemin indirect (multi-saut) reliant l'émetteur avec le sink où le nœud (N11) reste en mode actif. Les nœuds (N12, N14, N15) sont basculés en mode veille, car ils se trouvent dans le rayon de transmission de l'émetteur.

• **Quatrième expérimentation :**

Le tableau suivant représente les paramètres utilisés :

Les paramètres	La surface	Nombre de nœuds	Rayon de transmission	Rayon de détections
	900 m x 900 m	25	150 m	120 m

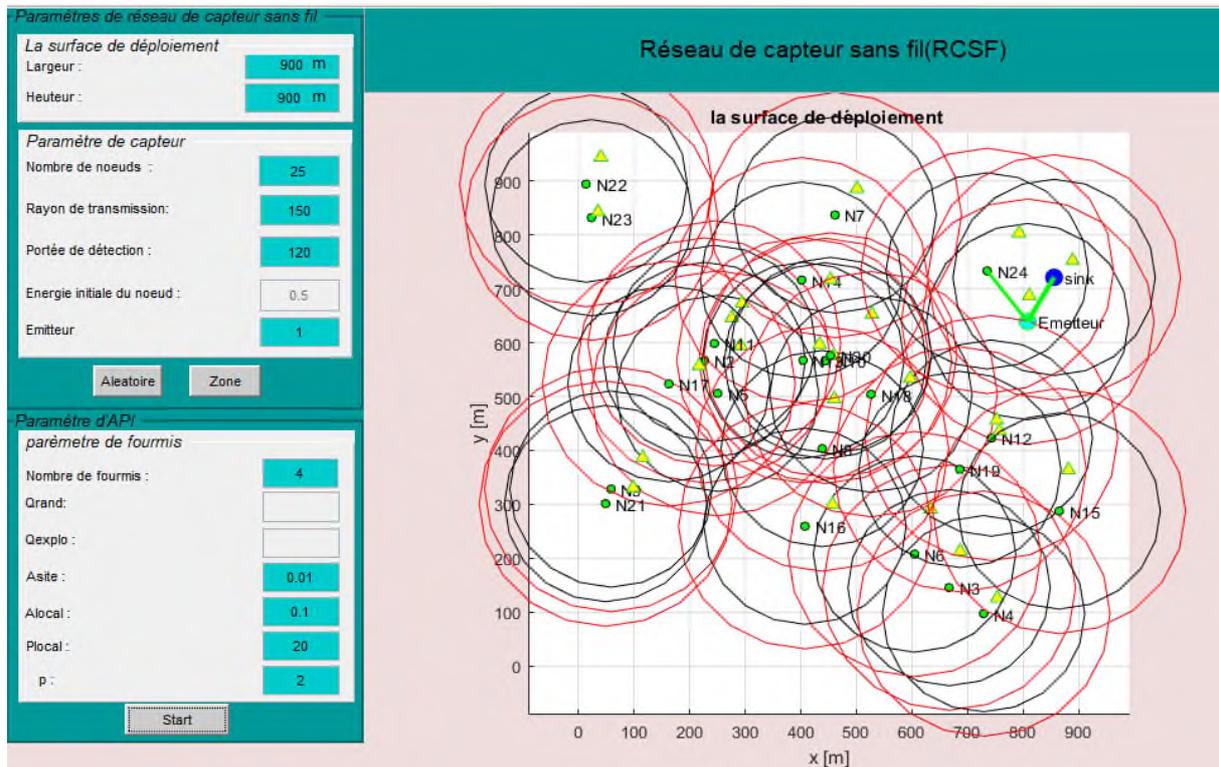


Figure 4.10 : La quatrième expérimentation avec 25 nœuds de capteur et 25 cibles déployés de manière aléatoire.

De la figure 4.10 et après l’exécution de l’algorithme d’API sur le réseau défini et avec le choix du premier nœud (N1) comme un nœud émetteur, on remarque que l’algorithme d’API à déterminer le plus court chemin direct (mono-saut) reliant l’émetteur avec le sink. Le nœud (N24) bascule en mode veille, car il se trouve dans le rayon de transmission de l’émetteur et de récepteur.

- **Cinquième expérimentation:**

Le tableau suivant représente les paramètres utilisés :

Les paramètres	La surface	Nombre de nœuds	Rayon de transmission	Rayon de détections
	1000 m x 1000 m	50	150 m	120 m

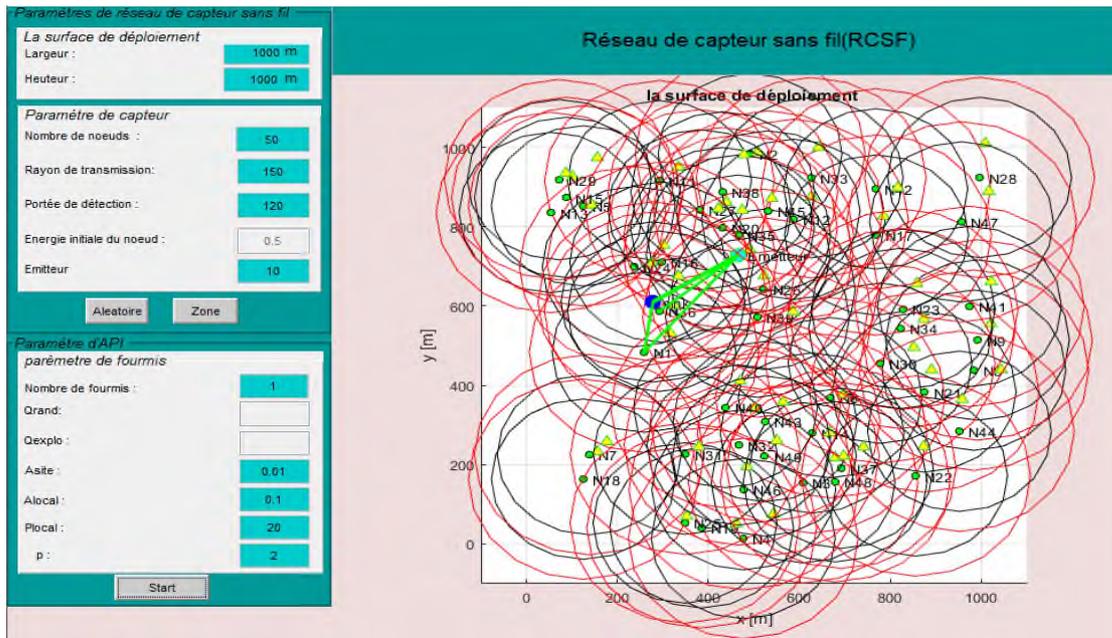


Figure 4.11 : La cinquième expérimentation avec 50 nœuds de capteur et 50 cibles déployés de manière aléatoire.

De la figure 4.11 et après l’exécution de l’algorithme d’API sur le réseau défini et avec le choix du dixième nœud (N10) comme un nœud émetteur, on remarque que l’algorithme d’API adapté à déterminer le plus court chemin indirect (multi-saut) reliant l’émetteur avec le sink. Le nœud (N36) bascule en mode veille, car il se trouve dans le rayon de transmission de l’émetteur.

4.7Analyse des résultats : Après plusieurs expérimentations faites on obtient le résultat suivant : Le tableau suivant représente les nœuds actifs et les nœuds désactivés :

Le nombre des nœuds	4	12	16	25	50
Les nœuds actifs	2	3	3	2	3
Les nœuds désactivés	1	1	3	1	1

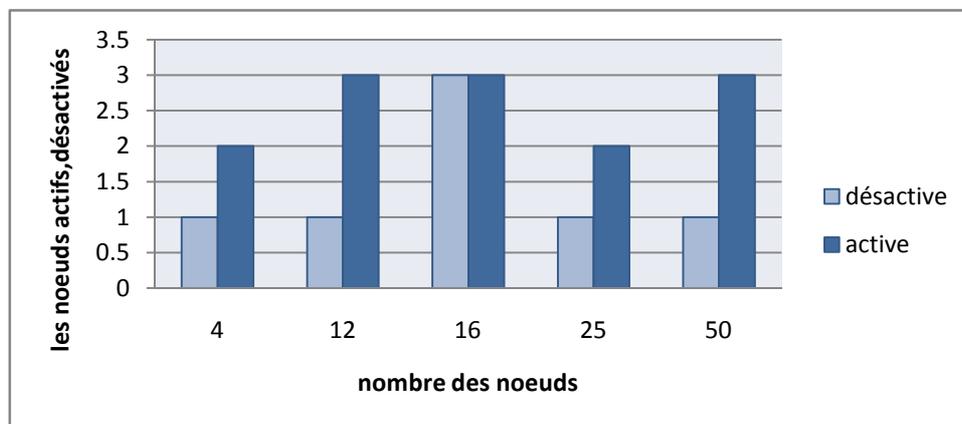


Figure 4.12: Un organigramme représentant les nœuds actifs les nœuds désactivés.

4.8 Conclusion :

Dans ce chapitre, nous avons décrit l'adaptation de l'algorithme d'API au problème de 1-couverture en calculant le plus court chemin entre le nœud source (émetteur) et la station de base. Cet algorithme adapté peut fournir une 1-couverture des points cibles et également maintenir la connectivité entre les nœuds de capteurs actifs et la station de base pour transmettre les données de détection dont la durée de vie du RCSF est améliorée en désactivant les nœuds redondants. En utilisant divers exemples de réseau, l'algorithme d'API adapté est largement simulé avec un nombre variable de nœuds de capteurs et de points cibles, ainsi que la densité du réseau.

Conclusion générale

Conclusion générale :

Les réseaux de capteurs sans fil (RCSF) constituent un axe de recherche très essentiel et peuvent être appliqués dans plusieurs domaines différents. Les RCSF sont principalement utilisés pour surveiller des cibles dans le domaine concerné. L'un des domaines de recherche les plus actifs dans les RCSF est celle de la couverture. La couverture et la connectivité sont deux des problèmes les plus initiaux du RCSF. La couverture dans les RCSF est généralement définie comme une mesure de la qualité de la surveillance du champ de détection ou de la capacité des capteurs à observer l'espace physique. La connectivité peut être définie comme la capacité du capteur à atteindre la station de base. Dans ce mémoire en proposant l'adaptation d'un algorithme de colonie de fourmis appelé API afin de traiter le problème de couverture (1-couverture) dans ce type de réseau.

Initialement, dans le premier chapitre de ce mémoire, nous avons débuté par une description générale sur les RCSF, puis nous avons défini les points essentiels relatifs aux structures de RCSF, ainsi que leurs caractéristiques et les domaines d'applications. Dans le deuxième chapitre, nous avons commencé par présenter de quelques travaux existant traitant le problème de la couverture dans les RCSF tels que le problème de 1-couverture et k-couverture, ensuite nous avons exposé les types et les paramètres de la couverture dans les RCSF. En fin, nous avons décrit les critères liés au problème de couverture dans les RCSF et la classification des protocoles de la couverture de surface. Dans le troisième chapitre, nous avons présenté quelques méta-heuristiques (les méthodes trajectoires et les méthodes de population). Et ensuite, nous avons détaillé l'algorithme de colonie de fourmis API (*Pachycondyla-apicalis*) que nous avons l'adapté pour traiter le problème de la couverture (1-couverture) de cibles dans notre travail.

Dans ce travail de mémoire, nous avons proposé l'adaptation de l'algorithme API au problème de 1-couverture. Cet algorithme adapté peut fournir une couverture complète des points cibles et également maintenir la connectivité entre les nœuds de capteurs active et la station de base pour transmettre les données de détection. Une fonction objectifs efficace est utilisée sous des contraintes, capable de gérer l'objectif de la couverture simple (1-couverture) traité avec l'amélioration de la durée de vie du RCSF en désactivant les nœuds redondant. En utilisant divers scénarios de réseau, l'algorithme d'API adapté proposé est largement simulé avec un nombre variable de nœuds de capteurs et de points cibles, ainsi que la densité du réseau. En fin nous pouvons dire que notre travail est extensible. De ce fait, plusieurs améliorations et perspectives futures restent à étudier. Entre elles, nous pouvons citer :

- ❑ Comparer notre travail de ce mémoire à d'autres travaux similaires ;
- ❑ Simuler l'adaptation de l'algorithme d'API pour la 1-couverture par d'autres simulateurs ;
- ❑ Adapter le même algorithme d'API pour le problème de k-couverture.....

Bibliographie

Bibliographie

- [1] O. Smail. « Routage multi path dans les réseaux Ad hoc ». Thèse de Doctorat, Université Mohamed Boudiaf USTO-MB Oran, 2015.
- [2] Nadjib Badache «réseaux de capteurs», rapport de recherche dans les systèmes informatiques, février 2004
- [3] Rahim KACIMI « Techniques de conservation d'énergie pour les réseaux de capteurs sans fil» Thèse de Doctorat, Université Toulouse Le 28 Septembre 2009
- [4] E. Fleury «réseaux de capteurs», 16 décembre 2004.
- [5] N.Labraoui, « la sécurité dans les réseaux de capteurs sans fil », thèse de doctorat, Université de Tlemcen, 2012.
- [6] M.E.Khanouche, « Le traitement du problème de la couverture dans les réseaux de capteurs sans fil », mémoire de magistère, Université de Bejaia, 2010.
- [7] C. Narimane et H. Med Abdenour « Couverture et connectivité dans un réseau de capteurs sans fil »
- [8] Diery Ngom. « Optimisation de la durée de vie dans les réseaux de capteurs sans fil sous contraintes de couverture et de connectivité réseau. Réseaux et télécommunications [cs.NI]. » Université de Haute Alsace - Mulhouse, 2016. Français
- [9] V.Rajavavivarme, Y.Yang, and T.Yang « An overview of wireless sensor network and applications. In Proceedings of the 35th Southeastern Symposium on System Theory» pages432436, 2003
- [10] P. Johnson and D.C Andrews « Remote continuous monitoring in the home. Journal of Telemedicine and Telecare», vol.2, no.2, pp.107-113, June 1996.
- [11] S. Maarouf – S. Ouadah « Implémentation et évaluation des schémas de routage sur une plateforme réelle de réseaux de capteurs sans fil » Université Abou Bakr Belkaid– Tlemcen 25 Juin 2014.
- [12] A.Giridhar and P.Kumar « Maximizing the functional lifetime of sensor networks, inproceedings of the 4th International Symposium on Information Proceeding in Sensor Network (IPSN) » , 2005, pp. 13-19.
- [13] Y. Chen and Q. Zhao «On the Lifetime of Wireless Sensor Networks, in IEEE Communication letters », Vol. 9, No. 11, November 2005, pp. 976-978.
- [14] M. Esseghir, N.Bouabdallah, and G.Pujolle «A Novel Approach for Improving Wireless Sensor Network Lifetime, in IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications», 2005, pp. 2701-2706.
- [15] Z. Hu and B. Li, On the fundamental capacity and lifetime limits of energy-constrained wireless sensor networks», in Proceedings of 10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004.
- [16] S.Moad, “Optimisation de la consommation d'énergie dans les réseaux de capteurs sans fil”, Master 2 de Recherche en Informatique Université Rennes, 2008, pp. 1-50.
- [17] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. Peas: A robust energy conserving protocol for long-lived sensor networks. In Proceedings of International Conference on

Bibliographie

- Distributed Computing Systems (ICDCS), pp. 28-37, Rhode Island, USA, 2003.]
- [18] 1319-1578 _ 2016 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).
- [19] C.F.Huang,Y.C.Tseng.Asurvey of solutions to the coverage problem sinwireess sensor networks. Journal of internet Technology, 6 (1) 1-8, 2005
- [20] M. Cardei and J. Wu. Energy-efficient coverage problems in wireless ad-hoc sensor networks. Elsevier Computer Communications Journal, 29 (4): 413-420, 2006.
- [21]S.SoroandW.B.Heinzelman.Clusterheadelectiontechniquesforcoveragepreservationinwirel esssensor networks.ElsevierAdHocNetworksJournal, 7(5):955-972, 2009.
- [22]S.Meguerdichian,F.Koushanfar,M.PotkonjakandM.B.Srivastava.CoverageProblem sin Wireless Ad-Hoc Sensor Networks.In20th Annual Joint Conference of Societies, 3: 1380-1387, 2001
- [23] C. GUI and P. Mohapatra. Power conservation and quality of surveillance in target tracking sensor networks. In Proceedingsofthe10thAnnual International Conference on Mobile Computing and Networking, pages 129–143, 2004.
- [24] S. K. Gaurav, B. Yigal and S.Saswati.Life time and Coverage Guarantees through Distributed Coordinate-Free Sensor Activation. In Proceedings of the 15thannual international conference on Mobile computing and networking, pages 169-180, 2009
- [25] J. Deng, Y. S. Han,W.B.HeinzelmananP.K.Varshney.Scheduling Sleeping Nodes in High Density Cluster-based Sensor Networks. Mobile Networks and Applications, 10(6): 825-835, 2005.
- [26] V.Tran-Quang and T.Miyoshi.Anovel gossiP-based sensing coverage algorithm for dense wireless sensor networks. Eesvier Computer Networks Journal, 53(23): 2275-2287, 2009
- [27] V.Raghunathan,A.Kansai, J.Hse, J.Friedmanand M.Srivastava.Design considerations for solar energy harvesting wireless embedded systems. In Proceedings of the 4thinternational symposium on Information processing in sensor networks, pages 457-462, 2005
- [28] I. Chlamtac, I. Carreras and HagenWoesner. From Internets to Bionets: Biological Kinetic Service Oriented Networks. In K. B.Szymanskiand B. Yener Advances in Pervasive Computing and Networking, pages 75-95, 2005.
- [29] J. Zhang, T. Yan, and S.H.Son,Deployment strategies for differentiated detection in wireless sensor networks,3rdAnnual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, SECON, vol.1, pp. 316–325, 2006
- [30] N. Aitsaadi, N. Achir, K. Boussetta and G. Pujolle, Differentiated Underwater Sensor Network Deployment, IEEE/OES OCEANS'07, 2007
- [31] G. Xing, C. Lu, R. Pless and J. A. O’Sullivan. Co Grid. an efficient coverage maintenance protocol for distributed sensor networks. In Proceedings of Information Processing in Sensor Networks, pages: 414-423, 2004
- [32] Z. Zhou, S. Das, and H. Gupta. Connected k-coverage problem in sensor networks. In proceedings of 13th International Conference on Computer Communications and Networks (ICCCN), pp.373-378, Chicago, IL, USA, 2004.
- [33] A. Gallais, J. Carle, D. Simplot-Ryl and I. Stojmenovic. Localized Sensor Area Coverage with Low Communication Overhead. In IEEE transactions on mobile computing, 7(5): 661-672, 2008.

Bibliographie

- [34] M.E.Khanouche, « Le traitement du problème de la couverture dans les réseaux de capteurs sans fil, mémoire de magistère », Université de Bejaia, 2010.
- [35] A. Khalil, « Méthodes analytiques pour la couverture dans un réseau de capteurs sans fil, mémoire de magistère », Université de Béjaïa, 2010.
- [36] Wang et al, articles universitaires correspondant aux termes « Solutions basées sur des méthodes géométriques », 2006.
- [37] Akkaya et al, articles universitaires correspondant aux termes « Heuristique distribuée et probabiliste de "clustering » », 2009.
- [38] Butenko et al, articles universitaires correspondant aux termes « Heuristique distribuée et probabiliste de "clustering » », 2003.
- [39] Iyengar et al ,2005
- [40] A. Ephremides, J. E. Wieselthier, and D.J. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. In Proceedings of the IEEE, vol.75, no.1, pp.56-73, January 1987.
- [41]A. Makhoul, R. Saadi, and C. Pham. Surveillance vidéo sur réseaux de capteurs sans-fils : ordonnancement adaptatif avec prise en compte de la criticité, Université de Pau et des Pays de l'Adour (UPPA) ,2009.
- [42] J.Dreo « Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues .Application en génie biomédical »
- [43]C.H.PapadimitriouandK.Steiglitz.CombinatorialOptimization:AlgorithmsandComplexity. Prentice-Hall, 1982.
- [44] S. Kirk Patrick, C. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. Science, 220:671–680, 1983.
- [45] V. Cerny. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm.J. Of Optimization Th. And applications, 45:41–51, 1985.
- [46] H. HACHIMI « HYBRIDATIONS D'ALGORITHMES MÉTAHEURISTIQUES EN OPTIMISATION GLOBALE ET LEURS APPLICATIONS » le 29 Juin 2013
- [47] Colomi, A., Dorigo M. et Maniezzo V., « Distributed optimization by ant colonies », dans : Toward a practice of autonomous systems : proceedings of the First European Conference on Artificial Life (ECAL91), VARELA F. J. et BOURGINE P. (éditeurs), MIT Press, Cambridge, 1992, 134–142.10. Atrick Siarry - Éric Taillard
- [48] C. Andrea , L. Thé Van et M.Guillaume « Optimisation par colonies de fourmis » 19 mai 2006.
- [49] Dorigo 03 « Un chapitre dans son livre qui parle des algorithmes élaborés des colonies de fourmis notamment adaptés pour le TSP et divers problèmes d'optimisation dynamiques »
- [50] N. Monmarché, Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation. Interface homme-machine [cs.HC]. Université François Rabelais, Tours, 2000.
- [51] N.Monmarché, « optimisation combinatoire (problème du voyageur de commerce) » 1998.
- [52] T. BROUARD, M. SLIMANE, J.-P. ASSELIN DE BEAUVILLE, G. VENTURINI Apprentissage d'une chaîne de Markov cachée »Problèmes numériques liés à l'application à l'image » Revue de statistique appliquée, tome 46, n°2 (1998), p. 83-108
- http://www.numdam.org/item?id=RSA_-1998_-46_-2_-83-0

Bibliographie

[53] Claude Touzet, les réseaux de neurones artificiels « introduction au connexionnisme cours, exercices et travaux pratiques », Juillet 1992

[54] Subash Harizan, Pratyay Kuila « Coverage and connectivity aware energy efficient scheduling in target based wireless sensor networks: an improved genetic algorithm based approach» Springer Science+Business Media, LLC, part of Springer Nature 2018