



Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Electronique et d'Informatique

Département d'Informatique

Mémoire de Master

Spécialité:

Réseau et Systèmes Distribués

Thème

La couverture basée transport dans l'architecture IoT

SUJET PROPOSÉ PAR :
Mme. BELGUERCHE Nadia

PRÉSENTÉ PAR :
AOUDJEGHOUT Hanene
DEMIGHA Hamza Amine

Projet RSD010/2021

Remerciements

La réalisation de ce mémoire a été possible grâce à l'aide de plusieurs personnes à qui nous tenons à exprimer notre gratitude.

*Nous tenons tout d'abord à adresser toute notre gratitude à Mme **BELGUERCHE** Nadia , notre encadreur, pour sa patience, sa disponibilité et surtout ses conseils judicieux, qui ont contribué à nourrir notre réflexion.*

*Nous tenons également à remercier les professeurs de l'université d '**USTHB** , qui nous ont fourni les outils nécessaires pour réussir nos études universitaires.*

Un grand merci à tous nos amis, qui nous ont apporté leur soutien moral et intellectuel tout au long de notre démarche.

Nous tenons à exprimer notre gratitude aux membres de jury pour avoir accepté de juger ce travail.

Enfin, nous tenons à exprimer notre gratitude à nos parents pour leur confiance et leur soutien inestimable.

Dédicaces

Je dédie ce mémoire à :

Ma mère, ma véritable partenaire pendant cinq années d'études, cette femme merveilleuse qui a œuvré pour ma réussite, par son amour, son soutien, ses prières, tous les sacrifices qu'elle a consentis et ses précieux conseils, pour toute son aide et sa présence dans ma vie.

*Mon père, qui peut être fier de trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Que Dieu fasse que ce travail porte ses fruits ;
Merci pour les nobles valeurs, l'éducation et le soutien permanent de votre part.*

A mes frères et mes sœurs.

Aoudjeghout Hanene

Je dédie ce mémoire à :

A ma mère, qui depuis ma naissance, n'a cessé de me bercer avec des bons conseils et sans doute celle qui me réconforte et m'encourage dans les moments les plus difficiles de ma scolarité.

A mon père, celui qui ne se lassait jamais pour aménager tous ses efforts pour que je sois quelqu'un de bien dans la vie. Pour lui, l'instruction est le socle fondamental de ses enfants.

A mon cher frère, qui, par son soutien moral et encouragement, multipliait mes efforts, pour pouvoir réaliser parfois l'impossible.

A mes amis, aussi à tous ceux qui m'ont soutenues par leurs orientations, leurs conseils durant la réalisation de ce travail. Je vous dédie ce travail avec tous mes vœux de bonheur, et de réussite.

Demigha Hamza Amine

Résumé

L'Internet des objets (IdO) désigne un type de réseau permettant de connecter n'importe quel objet sur la base de protocoles stipulés, par le biais d'équipements de détection d'informations, afin d'échanger des informations et de communiquer dans le but de réaliser des reconnaissances, des positionnements, des traçages, des contrôles et des administrations intelligents. L'une des questions les plus importantes de la recherche sur l'IdO est de savoir comment déployer efficacement les capteurs dans la ROI. Grâce au déploiement efficace de capteurs, les applications basées sur l'IdO peuvent recueillir suffisamment d'informations pour faciliter la prise de décision.

Dans ce présent mémoire, nous proposons une solution pour déplacer des capteurs redondants dans le réseau de capteurs et de robots sans fil (RCSRFs) afin de résoudre le problème de déploiement de capteurs dans la couche de perception de l'Internet des objets. Dans un premier temps, des capteurs seront déployés de manière aléatoire dans la zone d'intérêt. Par la suite, des capteurs redondants et des trous de couverture seront détectés, et des robots avec des charges de capteurs variables pourront réparer les trous pour augmenter la couverture de la zone en utilisant le principe de réflexion de la lumière.

Mots clés : IdO ,déploiement , robot , capteur , réflexion , couverture , trou, RCSRFs.

Abstract

The Internet of Things (IoT) refers to a network that connects everything with the Internet in accordance with prescribed protocols through information sensing equipment to exchange and communicate information to achieve intelligent identification, positioning, tracking, monitoring and management. One of the most important issues in IoT research is how to effectively deploy sensors in the target field. Through the effective deployment of sensors, IoT-based applications can collect enough information to support decision-making.

In this paper, we propose a solution to move redundant sensors in the Wireless Sensor and Robot Network (WSRNs) to solve the sensor deployment problem in the perception layer of the Internet of Things. Initially, sensors will be deployed randomly in the area of interest. Later, redundant sensors and coverage holes will be detected, and robots with varying sensor loads can repair the holes to increase the area coverage using the principle of light reflection.

Keywords: IoT , coverage , déploiement , robot , sensor , hole , reflection, WSRNs.

Table des matières

Introduction Générale	11
1 Réseaux de Capteurs sans Fil dans l'internet des objets	13
1.1 Introduction	14
1.2 Internet des objets	14
1.2.1 Définition	14
1.2.2 Architecture	14
1.3 Généralités sur les réseaux de capteurs sans fil	15
1.3.1 Caractéristique d'un capteur	16
1.3.2 Architecture d'un capteur	17
1.3.2.1 Architecture matérielle	17
1.3.2.2 Architecture logicielle	19
1.3.3 Définition de réseau de capteurs sans fil (RCSF)	19
1.3.4 Architecture d'un réseau de capteur sans fil	20
1.3.5 Domaine d'application des RCSFs	20
1.3.6 Propriétés et Contraintes des RCSFs	23
1.3.7 Problème de la couverture dans les RCSFs	24
1.3.7.1 Degré de couverture	24
1.3.7.2 Type de problème de couverture dans RCSF	24
1.4 Les réseaux de capteurs et de robots sans fil (RCRSF)	25
1.4.1 Définition d'un robot	25
1.4.2 Définition d'un réseau de capteurs et de robots sans fil (RCRSF) . .	25
1.4.3 Les apports des robots avec les RCSF	25
1.5 Conclusion	26
2 État de l'art	27
2.1 Introduction	28
2.2 Principes et Techniques	28
2.2.1 Modélisation de réseaux de capteur	28
2.2.2 Capteur redondant	28
2.2.3 Trou de couverture	29
2.3 Déploiement dans les RCSF	29
2.3.1 Déploiement déterministe	29
2.3.2 Déploiement aléatoire	30
2.4 Solutions Antérieurs	31
2.4.1 Solution déterministe	31

2.4.1.1	Approche basée sur l'optimisation par colonies de fourmis (ACO-Greedy)	31
2.4.2	L'auto déploiement	32
2.4.2.1	Approche basée sur l'optimisation par essaims de vers luisants (GSO)	32
2.4.2.2	Algorithme de déploiement de capteurs assisté par des mouvements localisés pour la détection et la réparation de trous (HEAL)	33
2.4.3	Déploiement aléatoire basé Transport	34
2.4.3.1	Based Sensor relocation (CBR)	34
2.4.3.2	Algorithme Carrier-Based Sensor Deployment Algorithm for Perception Layer in the IoT Architecture	36
2.5	Tableau comparatif des algorithmes de déploiement	37
2.6	Conclusion	37
3	Conception	38
3.1	Introduction	39
3.2	Motivation	39
3.3	Hypothèse	40
3.4	structure de données	40
3.4.1	structure de données des capteurs	40
3.4.1.1	Messages envoyés par le capteur	41
3.4.2	structure de données du Robot	41
3.4.2.1	Messages envoyés par le robot	42
3.5	Technique de base	42
3.5.1	la réflexion de la lumière	42
3.5.2	Loi des Grands Nombres	44
3.5.3	Relation entre la loi des grand nombre et la réflexion de la lumière	44
3.5.4	Angle de départ	46
3.6	Algorithme proposé	46
3.7	Principe de Fonctionnement	48
3.7.1	Algorithme des Capteurs	48
3.7.1.1	Détection de voisinage	48
3.7.1.2	Détection capteurs redondant	48
3.7.1.3	Détection de trou de couverture	50
3.7.2	Algorithme de Robot	51
3.7.2.1	Exploration de la zone par le robot	51
3.7.2.2	Réparation des trous	52
3.8	Version d'obstacle de notre solution	53
3.9	Conclusion	54
4	Simulation et évaluation des performances.	56
4.1	Introduction	57
4.2	Outils de développement	57
4.3	Environnement de la simulation	57
4.4	Critère de performances	58

4.5	Paramètre de simulation	58
4.6	Résultats et évaluation des performances	59
4.6.1	Impact de la variation de l'angle de départ sur le temps de relocalisation	59
4.6.2	Impact de la variation de Pas du robot sur le temps de relocalisation	60
4.6.3	Impact de la variation de la charge du robot sur le temps de relocalisation	62
4.6.4	Impact de la variation de la surface	63
4.6.5	Comparaison de temps de relocalisation requis pour atteindre un certain taux de couverture	65
4.6.6	Impact de la présence d'obstacle sur notre solution	66
4.6.6.1	Sur le temps de localisation	67
4.6.6.2	Sur le nombre moyen de mouvements de robot	68
4.7	Analyse Générale	68
4.8	Conclusion	69
Conclusion Générale		70
Bibliographie		71

Table des Figures

1.1	Architecture d'un IdO [3]	15
1.2	Quelques capteurs existants sur le marché [5]	16
1.3	Rayon de communication et de sensation d'un capteur	17
1.4	Architecture matérielle d'un capteur sans fil [9]	17
1.5	Architecture d'un réseau de capteurs sans fil[14]	20
1.6	Les domaines d'applications des RCSF [8]	21
1.7	Réseaux de capteur et de robot sans fil (RCRSF)[25]	25
2.1	Capteur 5 est un capteur redondant	28
2.2	Trou de Couverture	29
2.3	Classification des solutions de déploiement des capteurs	30
2.4	Un exemple du problème GCLC. (a)Avant le déploiement du nœud et (b) Après le déploiement du nœud)[33]	31
2.5	Illustration du mouvement du capteur[34]	33
2.6	Principe du processus de guérison [35]	34
2.7	la partition d'un système de grille d'un RCSF [28]	36
3.1	Principe de la réflexion de la lumière	43
3.2	Exemple de l'utilisation du principe de réflexion	43
3.3	La marche aléatoire du robot	44
3.4	l'impact du nombre de pas du robot sur le principe de réflexion	45
3.5	Exemple de réparation d'un trou	53
3.6	Exemple illustratif de la marche du robot en présence des obstacles	54
4.1	Exemple d'un environnement de simulation	58
4.2	Impact de la variation de l'angle sur le temps de relocalisation	60
4.3	Impact de la variation de Pas du robot sur le temps de relocalisation	61
4.4	Impact de la variation de charge du robot sur le temps de relocalisation	63
4.5	Impact de la variation de la surface sur le temps d'exécution	64
4.6	Impact de la variation de la surface sur le nombre moyen de pas du robot	64
4.7	le temps de relocalisation requis pour atteindre un certain taux de couverture pour les différents solutions	66
4.8	Comparaison du temps de relocalisation entre la présence ou pas d'un obstacle dans notre solution	67
4.9	Comparaison du nombre moyen de mouvement du robot entre la présence ou pas d'un obstacle dans notre solution	68

liste des Tableaux

2.1	Tableau comparatif des algorithmes de déploiement	37
4.1	Les paramètres de simulation	59
4.2	Simulation 1	59
4.3	Simulation 2	61
4.4	Simulation 3	62
4.5	Simulation 4	63
4.6	Simulation 5	65
4.7	Simulation 6	67

Introduction Générale

Depuis son lancement, l'Internet des objets (IdO) a reçu un solide soutien de la part des gouvernements et des instituts de recherche du monde entier et a obtenu des résultats remarquables. En tant que lien entre l'Internet des objets et le monde réel, la couche de perception de l'Internet des objets joue un rôle important. Sa fonction principale est d'utiliser divers dispositifs de détection pour échantillonner des données environnementales et distinguer l'état de l'appareil. Il traite les données via un accès collaboratif pour obtenir des informations utiles, puis les transmet à la couche réseau via des dispositifs d'accès tels que RCSF. Dans la couche de perception de l'Internet des objets, le réseau de capteurs sans fil (RCSF) joue un rôle essentiel. En effet, RCSF nous permet d'obtenir différents types d'informations sur l'environnement, faisant ainsi le lien entre le monde physique et le monde numérique.

La performance d'un réseau de capteurs sans fil est grandement influencée par le processus de déploiement des nœuds de capteurs. La question du déploiement et du positionnement des nœuds de capteurs dans un RCSF est une stratégie qui est utilisée pour définir la topologie du réseau, le nombre et la position des nœuds de capteurs.

Les différentes tâches au niveau du déploiement peuvent être rassemblées sous trois phases principales. Une phase de prédéploiement et de déploiement concerne le placement manuel des nœuds par un humain ou un robot, ou le lancement des nœuds à partir d'un hélicoptère par exemple. On y distingue alors deux types de déploiement : un déploiement déterministe, plaçant chaque nœud dans une position connue préalablement, et un déploiement non déterministe, disposant aléatoirement les nœuds en les parachutant d'un avion ou hélicoptère, engendrant ainsi des trous de couverture (sous-région non couverte) et des capteurs redondants (sous-région couverte par plusieurs capteurs). Une deuxième phase qui est la phase de post-déploiement, elle est nécessaire si la topologie du réseau a évolué, suite par exemple à un déplacement de nœuds, ou un changement des conditions de propagation radio. La troisième phase considère le redéploiement qui consiste à ajouter de nouveaux nœuds au réseau pour remplacer certains nœuds défectueux ou pour relocaliser les capteurs redondants afin de couvrir les trous de couverture.

Dans le but de maximiser la couverture d'une région d'intérêt qui est parmi les problèmes les plus prépondérants pour garantir la qualité de service dans un réseau sans fil, deux méthodes ont été introduites. La première méthode consiste à disperser des capteurs mobiles et à laisser les capteurs se déplacer vers les emplacements désignés par le biais de la communication et de la coordination inter-capteurs. La deuxième méthode consiste à utiliser un robot mobile comme véhicule pour transporter des capteurs statiques et les déposer à des endroits précis et c'est le cas de notre solution.

Ce mémoire est structuré autour de quatre chapitres suivis d'une conclusion générale. La démarche de notre travail est présentée dans les deux premiers chapitres, tandis que notre apport sera détaillé dans le chapitre trois.

Le premier chapitre décrit de manière générale le contexte sur l'internet des objets et les réseaux de capteurs ainsi que le concept des robots mobiles.

Le deuxième chapitre présente un état de l'art sur les différents algorithmes de déploiement.

Le troisième chapitre est consacré à notre solution que nous avons proposées pour résoudre les problèmes de couverture en utilisant un processus de relocalisation assisté par un robot.

Le quatrième chapitre va aborder les différents tests pratiques réalisés permettant une évaluation aussi précise que possible à travers différents scénarios.

Chapitre 1

Réseaux de Capteurs sans Fil dans l'internet des objets

1.1 Introduction

L'internet des objets (IdO) est défini comme un paradigme dans lequel des objets équipés de capteurs, d'actionneurs et de processeurs communiquent entre eux pour servir un objectif utile. Ces objets possèdent des identifiants uniques et peuvent communiquer entre eux par une liaison sans fil. L'architecture IdO comporte trois couches, à savoir la couche de perception, la couche réseau et la couche application. La principale tâche de la couche de perception est de percevoir l'environnement, dans laquelle les réseaux de capteurs sans fil (RCSF) jouent un rôle crucial. En effet, les RCSFs nous permettent d'acquérir différents types d'informations sur l'environnement et, par conséquent, de jouer le rôle d'interface entre l'utilisateur et le réseau.

Dans le premier chapitre, nous avons d'abord présenté l'Internet des objets, les réseaux de capteurs, leur architecture et leurs caractéristiques. Ensuite, nous discutons du domaine d'application et des contraintes et problèmes de couverture dans RCSF. Enfin, nous avons montré l'utilité des robots dans ce type de réseau.

1.2 Internet des objets

1.2.1 Définition

Dans l'internet des objets (IdO), un objet peut être une personne équipée d'un moniteur cardiaque, un animal d'élevage doté d'un transpondeur à puce biologique, une automobile dotée de capteurs intégrés pour alerter le conducteur lorsque la pression des pneus est faible ou tout autre objet naturel ou artificiel auquel on peut attribuer une adresse IP (Internet Protocol) et qui peut transmettre des données sur le réseau.

De plus en plus, les organisations de divers secteurs utilisent l'IdO pour fonctionner plus efficacement, mieux comprendre les clients afin d'offrir un meilleur service à la clientèle, améliorer la prise de décision et accroître la valeur de l'entreprise[1].

1.2.2 Architecture

L'architecture de l'internet des objets est un cadre qui spécifie les éléments physiques à utiliser, les dispositions et paramètres techniques du réseau, les procédures d'exploitation et les formats de données. L'architecture à trois niveaux[2] est une structure courante et bien connue comme le montre la **Figure 1.1**.

- **La couche d'application** : Est la couche qui fournit des services d'information conscients pour les téléphones mobiles, les PC et autres dispositifs terminaux .
- **la couche transport** : Les données obtenues par les appareils doivent être diffusées et stockées. C'est la responsabilité de cette couche. Elle est responsable de la

transmission des données, de la connexion des objets intelligents, des périphériques réseau et des serveurs. Elle est aussi utilisée pour distribuer et analyser les données des capteurs.

- **La couche de perception** : Est la couche physique de l'architecture IoT. On y utilise principalement des capteurs et des systèmes embarqués. Ceux-ci collectent de grandes quantités de données en fonction des besoins. Elle comprend également des périphériques, des capteurs et des actionneurs qui communiquent avec l'environnement. Elles détectent certains paramètres spatiaux ou d'autres choses/objets intelligents dans l'environnement.

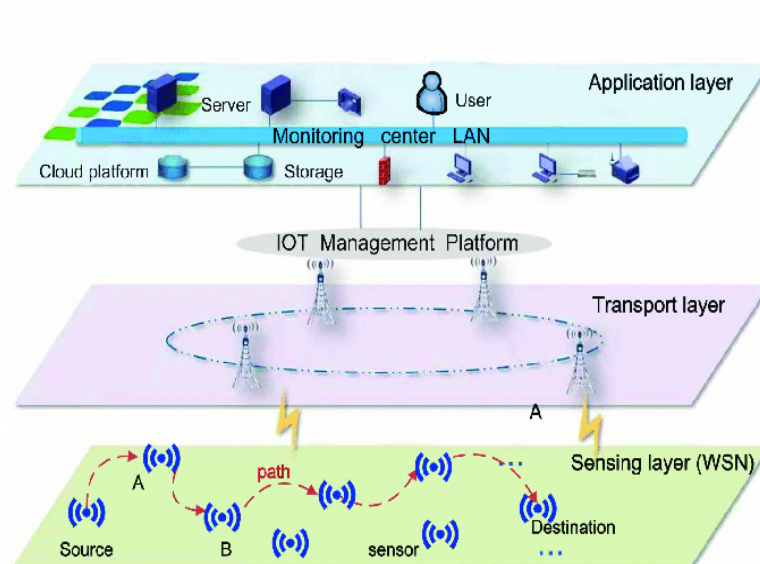


Figure 1.1: Architecture d'un IdO [3]

1.3 Généralités sur les réseaux de capteurs sans fil

Les capteurs sont des dispositifs miniaturisés avec des ressources énergétiques limitées et autonome, capable de traiter l'information et de la transmettre par ondes radio [4]. Parmi ces phénomènes collectés, on peut citer les phénomènes liés à la température, l'humidité, la pression, la capture d'images, etc

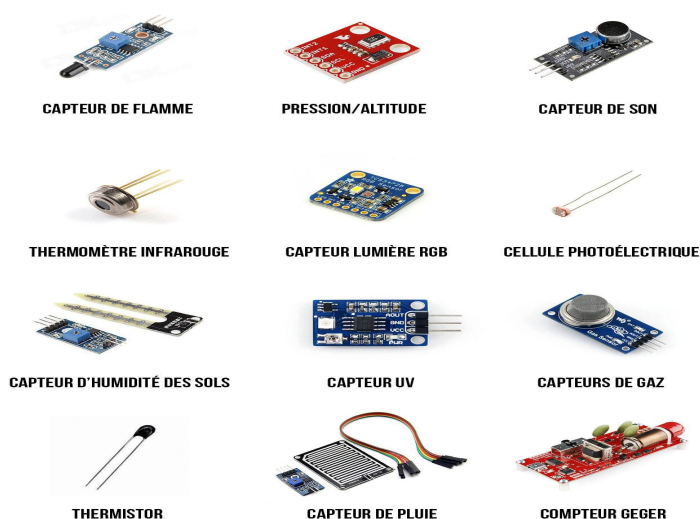


Figure 1.2: Quelques capteurs existants sur le marché [5]

les capteurs intègrent : une unité de capteurs chargés de collecter des grandeurs physiques (température ,humidité , vibration) et de les transformer en grandeur numérique, une unité de traitement informatique et de stockage de données et transmission sans fil.

1.3.1 Caractéristique d'un capteur

Les RCSFs présentent des caractéristiques propres au niveau des capteurs du réseau comme par exemple l'énergie, la portée de transmission, ainsi que de la puissance de stockage et au niveau du réseau qui est formé par ces nœuds comme par exemple la bande passante, le déploiement sur une surface précise, et la topologie du réseau. Les capteurs peuvent recueillir et transmettre des données captées dans l'environnement ou ils sont déployés. Le fonctionnement d'un capteur est basé sur deux entités fondamentales, l'unité d'acquisition qui représente le cœur physique permettant la prise de mesure et l'unité de communication qui réalise le transfert de celle-ci, vers d'autres dispositifs électroniques. Ainsi fonctionnellement, chaque capteur possède un rayon de communication (R_c) et un rayon de sensation (R_s) [6], comme le montre la **Figure 1.3** :

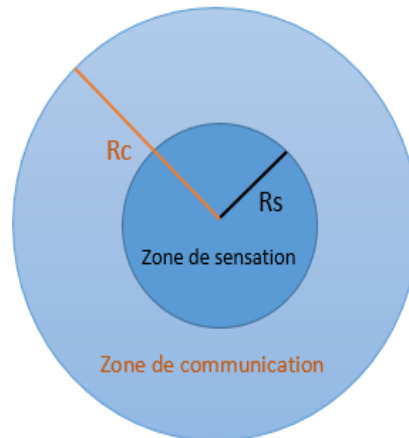


Figure 1.3: Rayon de communication et de sensation d'un capteur

1.3.2 Architecture d'un capteur

Concernant l'architecture des capteurs on peut la scinder en deux parties [7], ces deux parties à savoir l'architecture matérielle et l'architecture logicielle sont complémentaires car elles composent le capteur dans son intégralité, à cet effet nous développerons les deux architectures.

1.3.2.1 Architecture matérielle

Un capteur se compose de quatre unités de base : l'unité de capture, l'unité de traitement, l'unité de communication, et l'unité de contrôle d'énergie [8] qui se schématisent dans la **Figure 1.4** :

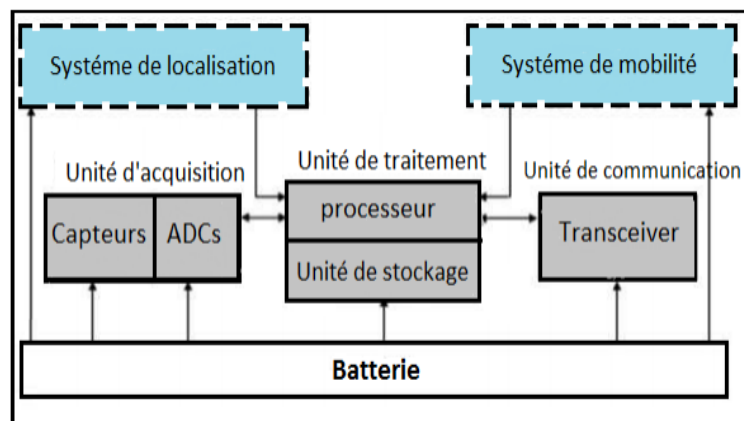


Figure 1.4: Architecture matérielle d'un capteur sans fil [9]

- **Unité de capture :**

On retrouve donc des équipements de différents types de détecteur et d'autre entrée. Le capteur est généralement composé de deux sous-ensembles : un récepteur (analyse d'identification) et un transducteur (convertissant le signal du récepteur en un signal électrique). Le capteur est chargé de fournir un signal analogique au convertisseur analogique/numérique en fonction du phénomène observé. Ce dernier convertit ces signaux en signaux numériques que l'unité de traitement peut comprendre.

- **Unité de traitement :**

L'unité de traitement est généralement composée de nœuds, de processeurs, de RAM et de mémoire flash. Une carte physique fonctionnant avec un système d'exploitation est généralement appelée un mote. Son noyau est un bloc composé d'un processeur, d'une RAM et d'une mémoire flash. Cette collecte est la base du calcul et du stockage binaire, une collecte temporaire de données et la collecte finale du système d'exploitation. Cette unité est chargée de mettre en œuvre des protocoles de communication qui permettent aux nœuds de coopérer avec d'autres entités du réseau. Elle peut également analyser les données capturées pour réduire la charge sur les nœuds récepteurs.

- **Unité de communication**

L'unité de communication est responsable de la transmission et de la réception de toutes les données via le support de communication radio. Elle peut-être de type optique ou de type radio-fréquence. Les communication de type optique son robustes vis-à-vis des interférences électrique. Cependant, son inconvénient est qu'il nécessite une ligne de vue permanente entre les entités communicantes. Par conséquent, elles ne peuvent pas établir de liaison à travers des obstacles .Bien que la consommation d'énergie de la communication radio-fréquence soit plus élevée, ce mode de communication reste le mode préféré pour la plupart des projets de recherche sur les réseaux de capteurs en raison de la petite taille des paquets de données échangés dans ces réseaux, de la faible vitesse de transmission et de la possibilité de réutilisation des fréquences. Il est également assez grand car la distance entre les nœuds est petite.

- **Unité de contrôle d'énergie**

Comme le montre la **Figure 1.4**, les trois unités précédent sont alimentés par une ressource énergétique (généralement une batterie) ,en conséquence de sa taille réduite la source énergétique dont il dispose est limitée et généralement irremplaçable, ce qui influe directement sur la durée de vie du réseau entier. Cette unité est responsable de répartir l'énergie disponible aux autres module et de réduire les dépenses en mettant en veille les composants inactif par exemple. Elle peut aussi gérer des systèmes de rechargement d'énergie à partir de l'environnement observé telles que les cellules solaires afin d'éteindre la durée de vie totale du réseau[10].

Il existe des capteurs qui sont dotés d'autres composants additionnels tels que :

- 1) **Système de localisation GPS (Global Position System)** : Il fournit des informations sur la localisation requise par les techniques de routage.
- 2) **Unité de mobilité (Mobiliser)** : Il est appelé si le nœud capteur doit être déplacé pour accomplir la requête à traiter.
- 3) **Module de mesure** : il mesure la distance ou l'angle avec un capteur voisin.

1.3.2.2 Architecture logicielle

Afin qu'un capteur puisse récolter et transmettre des données environnementales, il doit disposer de cinq sous-systèmes software de base :

- **Le système d'exploitation**: Le système d'exploitation open source TinyOS [11], basé sur le langage NesC , est conçu spécialement pour les capteurs sans fil. Ce système respecte une architecture reposant sur une association de composants. La bibliothèque de composants de TinyOS est particulièrement complète puisqu'elle inclut des protocoles réseaux, des pilotes de capteurs et des outils d'acquisition de données. En s'appuyant sur un fonctionnement événementiel, TinyOS propose à l'utilisateur une gestion très précise de la consommation du capteur et permet de mieux s'adapter à la nature aléatoire de la communication sans fil entre les interfaces physiques.
- **les drivers de perception** : Ce sont les modules software qui contrôlent les fonctions de base de perception du capteur. Les configurations appropriées doivent être chargées dans le capteur [11].
- **les drivers de communication** : Ce sont des modules software qui contrôlent la synchronisation, l'encodage du signal, la modulation. Ils sont implémentés dans la couche physique [11].
- **Processeur de communication** : Il contrôle les fonctions de communication: le routage, la maintenance de topologie, l'accès au média radio, décryptage... [11].
- **Intergiciel de traitement de données** : Il englobe le traitement des données et toutes les applications de base qui sont supportées au niveau du nœud (perception, sauvegarde de donnée...) [11].

1.3.3 Définition de réseau de capteurs sans fil (RCSF)

Un RCSF est un type particulier de réseau auto-organisé défini par un groupe de capteurs collaboratifs déployés dans une zone géographique appelée bassin versant ou zone

d'intérêt pour surveiller tout phénomène et collecter des données de manière autonome. Le capteur utilise une communication sans fil (Zig Bee ou Wifi) pour acheminer les données capturées via un routage multi-sauts vers un nœud considéré comme un "point de collecte", appelé station de base ou nœud récepteur. Ce dernier peut être connecté à des machines puissantes via Internet, réseau GPRS ou satellite. L'utilisateur peut envoyer sa demande au capteur en précisant les informations qui l'intéressent [12,13].

1.3.4 Architecture d'un réseau de capteur sans fil

Le réseau de capteurs sans fil se compose d'un grand nombre de nœuds de capteurs dispersés sur le terrain d'intérêt appelé « sites de puits ». Les nœuds ont la capacité de collecter périodiquement des données sur les phénomènes surveillés et d'envoyer des rapports de capture à un nœud spécial appelé puits (Sink), qui est le point de collecte des données capturées. Il peut transmettre les données collectées à l'utilisateur final via un réseau de communication (éventuellement Internet). L'utilisateur peut à son tour utiliser le récepteur comme passerelle pour transmettre sa requête au réseau. Afin de mieux comprendre le réseau de ces capteurs sans fil, la **Figure 1.5** montre leur mode de fonctionnement global.

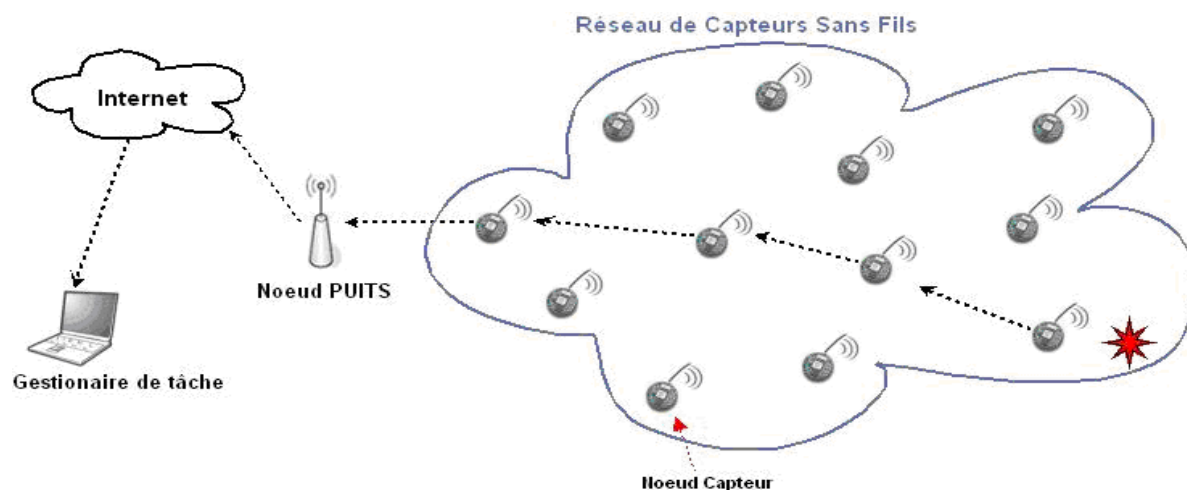


Figure 1.5: Architecture d'un réseau de capteurs sans fil[14]

1.3.5 Domaine d'application des RCSFs

La miniaturisation des capteurs, le coût de plus en plus faible, la large gamme des types de capteurs disponibles ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs de se développer dans plusieurs domaines d'application, Ils permettent aussi d'étendre les applications existantes. Les réseaux de capteurs peuvent se révéler très utiles dans de nombreuses applications lorsqu'il s'agit de collecter et de traiter des informations provenant de l'environnement. Parmi les domaines où ces

réseaux peuvent offrir les meilleures contributions, nous citons les domaines : militaire, surveillance, environnemental, médical, domestique, commercial, etc. Comme le montre les figures suivantes [8] :



Figure 1.6: Les domaines d'applications des RCSF [8]

- **Application militaire**

L'une des principales applications des réseaux de capteurs est le développement militaire. Dans ce cas, l'utilisation de réseaux de capteurs peut aller de la surveillance périphérique régulière à l'assistance aux attaques aériennes ou terrestres et à l'espionnage. A titre d'exemple d'applications dans ce domaine, on peut considérer des réseaux de capteurs déployés dans des endroits stratégiques ou difficiles d'accès pour surveiller toutes les activités des forces ennemies ou analyser le terrain avant de les y envoyer [15] . Étant donné que ces réseaux reposent sur le déploiement intensif d'un grand nombre de capteurs jetables et à faible coût, la destruction de certains capteurs n'affectera pas les actions des entreprises militaires, faisant du concept de réseaux de capteurs une meilleure méthode pour le champ de bataille.

De plus, il existe d'autres applications militaires auxquelles les réseaux de capteurs peuvent être appliqués :

- Le contrôle des forces, équipement et munition.
- Reconnaissance et surveillance du champ de bataille.
- Ciblage.
- Estimation des dégâts... etc.

- **Application médicale**

L'utilisation des réseaux de capteurs sans fil dans les applications médicales a suscité une attention particulière de la part des chercheurs. Il existe des micro-capteurs qui pourraient être avalés (sous forme de gélule) ou implantés sous la peau. Ces capteurs peuvent aider les médecins à recueillir des mesures sur certains paramètres physiologiques et à faire le suivi en temps réel des fonctions vitales de l'être humain. D'autres applications biomédicales ambitieuses basées sur les RCSFs existent également aujourd'hui, citons la surveillance de la glycémie, la surveillance à distance des soins pour lesquels la présence constante d'un professionnel de la santé n'est pas nécessaire, telle que la surveillance de la réadaptation, la surveillance des personnes âgées ou pour apporter une aide à une personne handicapée physiquement. À l'avenir, que ce soit à domicile, à l'hôpital ou ailleurs, les patients pourront être surveillés en permanence et les médicaments nécessaires leur seront fournis[11].

- **Application de surveillance**

L'application de réseaux de capteurs dans le domaine de la sécurité peut réduire considérablement les dépenses financières pour la protection des lieux et du personnel. Par conséquent, l'intégration de capteurs dans de grandes structures telles que des ponts ou des bâtiments aidera à détecter les fissures et les changements dans la structure en réponse aux tremblements de terre ou au vieillissement structurel. Le déploiement d'un réseau de capteurs de détection de mouvement peut fournir un système d'alarme pour détecter une intrusion dans la zone de surveillance [13].

- **Application industrielle**

Dans le domaine industriel, des capteurs ou des actionneurs peuvent être utilisés pour surveiller et contrôler les processus. Par exemple, dans une usine de traitement à plusieurs étapes, il peut y avoir des capteurs placés en différents points dans le processus afin de surveiller par exemple la température, la concentration chimique, et la pression [16].

- **Application environnementales**

Les réseaux de capteurs peuvent être utilisés pour surveiller les changements environnementaux. Ils permettent de déterminer la valeur de certains paramètres à un endroit donné, comme la température, la pression atmosphérique, etc. En dispersant des nœuds de capteurs dans le monde naturel, des événements tels que des incendies de forêt, des tempêtes ou des inondations peuvent être détectés. Cela permet une intervention plus rapide et plus efficace dans les services d'urgence. En agriculture, des capteurs peuvent être utilisés pour répondre de manière appropriée au changement climatique, par exemple en déclenchant le processus d'arrosage lorsqu'une zone sèche de terres agricoles est détectée [12].

1.3.6 Propriétés et Contraintes des RCSFs

La conception et l'implémentation des réseaux de capteurs sans fil sont guidées par plusieurs propriétés considérées comme des métriques de comparaison de performances entre les différents travaux dans le domaine.

- **Contrainte de couverture**

L'une des limitations les plus importantes de RCSF est qu'il fournit toujours une couverture fiable et complète d'un champ de détection spécifique et qu'il relaie toutes les données détectées au nœud de réception via un sous-ensemble des capteurs déployés [17].

- **Passage à l'échelle**

Le passage à l'échelle est une contrainte liée étroitement à la nature de l'application des réseaux de capteurs sans fil. Selon l'application, le nombre de nœuds déployés pour une zone d'intérêt peut atteindre plusieurs centaines voire des milliers de nœuds. Cependant, les solutions proposées doivent assurer l'efficacité et le bon fonctionnement des réseaux fortement denses[18].

- **Topologie dynamique**

Dans certaines applications, le réseau de capteurs sans fil doit supporter des topologies denses et dynamiques. La topologie d'un réseau de capteurs devient dynamique lorsqu'un nouveau nœud rejoint le réseau ou quitte le réseau à cause d'une défaillance. La topologie peut également changer dans le cas où les capteurs sont mobiles.

- **Tolérance aux pannes**

La défaillance de certains nœuds capteurs dans un réseau de capteurs sans fil peut être causée par plusieurs facteurs, notamment l'épuisement d'énergie, l'endommagement physique, ou aussi à cause des interférences liées à l'environnement. La tolérance aux pannes désigne l'habilité du réseau à maintenir ses fonctionnalités malgré les interruptions provoquées par la panne des capteurs. Elle vise à minimiser l'influence de ces défaillances sur le fonctionnement global du réseau [19]

- **Ressources limitées**

Les ressources des nœuds capteurs sont très limitées en ce qui concerne la puissance de calcul et la capacité de stockage, en plus de la ressource d'énergie électrique présentée ci-dessus. Certes l'efficacité en consommation d'énergie représente une métrique de performance significative mais si le nœud capteur effectue des traitements qui nécessitent un calcul complexe cela peut avoir un impact direct sur la durée de vie du réseau. Réduire la consommation de ressources tout en fournissant une qualité de service acceptable permet de prolonger la durée de vie du réseau[19].

1.3.7 Problème de la couverture dans les RCSFs

Le problème de couverture s'articule autour d'une question de base : « Comment plusieurs capteurs surveillent-ils une zone d'intérêt donnée ? ». La couverture est une mesure de la qualité du service (QoS) de la fonction de surveillance fournie par un réseau de capteurs. Le but est d'avoir tout les points physiques de la zone d'intérêts dans le rayon de capture d'au moins un capteur [20].

1.3.7.1 Degré de couverture

Selon les exigences et l'importance des applications implantées dans RCSF, le degré (niveau) de couverture peut se diviser en deux types: couverture simple et couverture multiple.

- **Couverture simple** : Dans les applications les moins sensibles telles que la surveillance des champs agricoles, nous pouvons concevoir des protocoles de couverture tels que chaque point dans la zone d'intérêt soit surveillé par un seul capteur, dans ce cas on parle de la 1-couverture ou de couverture simple [21].
- **Couverture multiple** : Dans les applications sensibles telles que les applications militaires ou liées à la sécurité, il est nécessaire d'assurer la couverture de chaque point dans la zone d'intérêt par plus d'un capteur pour permettre la tolérance aux fautes. Dans ce cas, on parle de la k-couverture ou de couverture multiple [21].

1.3.7.2 Type de problème de couverture dans RCSF

A la base de la nature de l'objet à surveiller, on distingue trois classes de couverture dans les réseaux de capteur , ce sont :

- **Couverture de zone** : s'appelle aussi la couverture de surface où l'objectif principal du réseau est de surveiller une région donnée, c.à.d la zone d'intérêt représente comme une région tel que tout point de cette région est couvert par au moins un capteur.
- **Couverture de point** : s'appelle aussi couverture de cible .dans ce type de couverture, on déploie des capteurs pour surveiller un ensemble de points spécifiques (cibles), dont la position géographique est connue.
- **Couverture de frontière** : pour couvrir une partie de la zone d'intérêt (région protégée), il n'est pas nécessaire d'avoir tous les points de la zone du déploiement couverts, on a besoin seulement de couvrir le périmètre de cette région pour diminuer des intrus par exemple.

1.4 Les réseaux de capteurs et de robots sans fil (RCRSF)

1.4.1 Définition d'un robot

Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) conçu pour accomplir automatiquement des tâches imitant ou reproduisant, dans un domaine précis, des actions humaines. La conception de ces systèmes est l'objet d'une discipline scientifique, branche de l'automatisme nommé robotique [22].

1.4.2 Définition d'un réseau de capteurs et de robots sans fil (RCRSF)

Le réseau de capteurs et de robots sans fil (RCRSF) est l'intégration du réseau de capteurs sans fil (RCSF) et du système multi-robot. Composés de capteurs en réseau et de robots mobiles, ils sont flexibles et ont moins de restrictions que les capteurs. Réaliser des tâches de détection et d'activation distribuées dans la région d'intérêt (region of interest- ROI). Le capteur peut également signaler au robot mobile que les capteurs à proximité ont échoué ou une couverture insuffisante de certaines zones à un robot mobile proche, il peut également collecter et de rapporter des données[23][24].

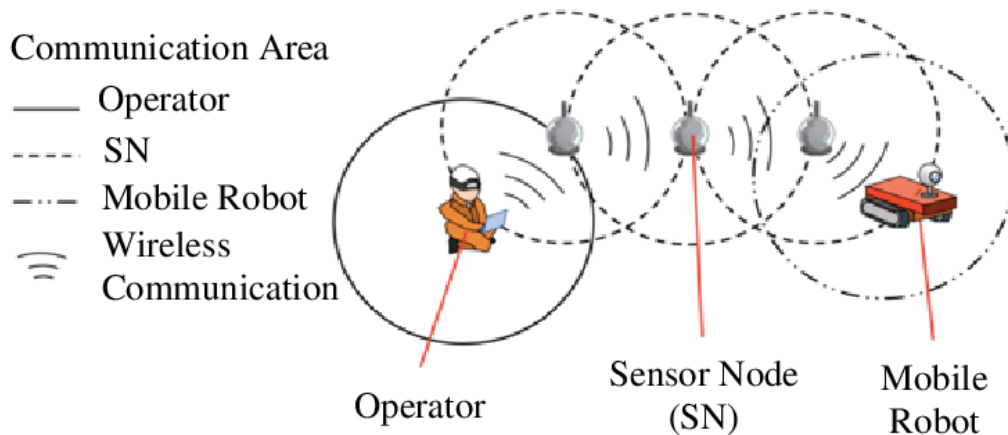


Figure 1.7: Réseaux de capteur et de robot sans fil (RCRSF)[25]

1.4.3 Les apports des robots avec les RCSF

Dans les réseaux de capteurs sans fil, les robots peuvent jouer un rôle essentiel. Ils peuvent être utilisés pour :

- **Remplacer des capteurs défaillants** : Détecter et trouver les nœuds défaillants et les remplacer par des nouveaux pour maintenir la connectivité du réseau[26].
- **Repositionner des capteurs** : Les robots peuvent être utilisés pour déplacer des capteurs fixes et statiques au lieu d'utiliser des capteurs mobiles[27][28].
- **Recharger des batteries** : Les robots alimentent les batteries des capteurs détectés qui ont un niveau d'énergie faible en déplaçant dans la zone[26].
- **Localiser les capteurs du réseau** : Le robot permet de connaître les positions des capteurs en recevant leurs signaux en déplaçant grâce à sa position qui'est connue à tout moment.[26].

1.5 Conclusion

Les réseaux de capteurs sans fil de l'Internet des objets (IdO) suscitent un intérêt considérable et représentent une nouvelle étape dans l'évolution des technologies de l'information et de la communication. Dans ce premier chapitre, nous avons essayé d'expliquer l'intérêt de RCSF pour l'IdO ainsi que leur architecture, contraintes de conception et leur problème de couverture.

Dans le prochain chapitre, nous discuterons l'état de l'art et de quelques algorithmes de couverture dans les RCSFs.

Chapitre 2

État de l'art

2.1 Introduction

La performance d'un réseau de capteur sans fil est considérablement influencée par le processus de déploiement des nœuds capteurs. La problématique de déploiement ou de positionnement des nœuds capteurs dans un RCSF est une stratégie qui sert à définir la topologie du réseau, donc le nombre et la position des nœuds capteurs. La qualité de la surveillance, la connectivité, et la consommation d'énergie sont aussi directement affectées par la topologie de réseau [29]. Dans ce chapitre, nous présenterons un état de l'art sur quelques travaux connexes sur le déploiement des capteurs dans les réseaux de capteurs sans fils.

2.2 Principes et Techniques

2.2.1 Modélisation de réseaux de capteur

Un réseau de capteur sans fils est représenté par un graphe $G = (V, E)$, avec V l'ensemble des sommets (les nœuds du réseau) et E l'ensemble des arêtes donnant les communications possibles.

2.2.2 Capteur redondant

Le capteur redondant fait référence au capteur qui couvre une sous zone déjà couverte par ses voisins, et qui peut être supprimé du réseau sans affecter le processus de réception des données ciblées [30].

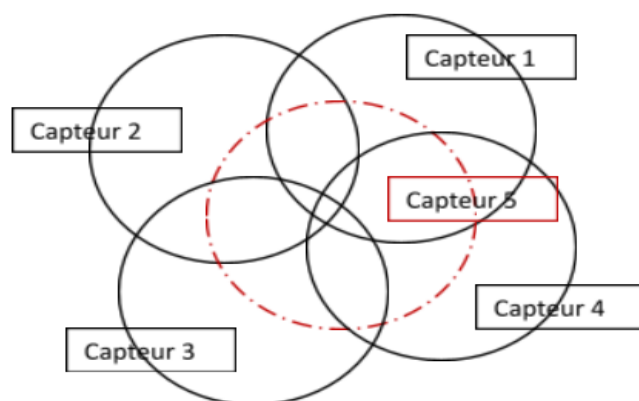


Figure 2.1: Capteur 5 est un capteur redondant

2.2.3 Trou de couverture

Un trou de couverture dans un réseau de capteurs peut être défini comme la zone du réseau qui n'est couverte par aucun nœud de capteur comme le montre la **Figure 2.2**, ce qui entraîne un manque de surveillance de cette zone.

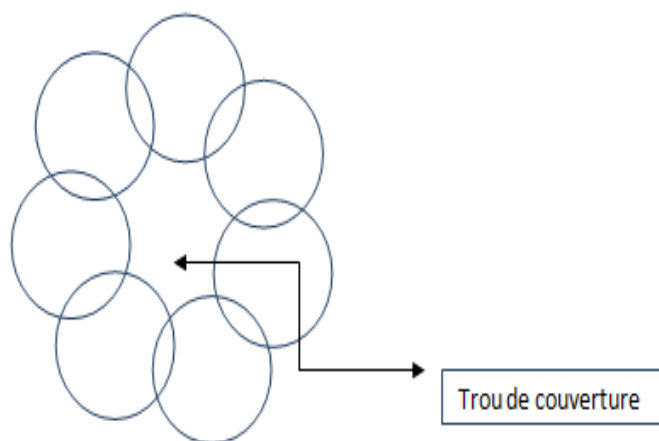


Figure 2.2: Trou de Couverture

2.3 Déploiement dans les RCSF

Dans les RCSF, les principaux défis sont le déploiement des nœuds capteurs dans la région d'intérêt pour satisfaire une détection incessante avec une durée de vie étendue du réseau et une zone de couverture uniforme. Plusieurs architectures et méthodes de déploiement des nœuds ont été développées pour les RCSF, en fonction des besoins de l'application. Le déploiement des capteurs peut être classé en fonction de la stratégie de placement, en déploiement aléatoire et déterministe[31].

2.3.1 Déploiement déterministe

La méthode de déploiement déterministe est adoptée principalement dans les cas où les emplacements des capteurs ont un grand impact sur le fonctionnement du RCSF, parce qu'il choisit le meilleur emplacement pour les capteurs, et leur position ne change pas pendant toute la durée de vie des RCSF, par exemple le déploiement de capteurs sous-marins,. En outre, cette méthode garantit une couverture maximale grâce à un déploiement optimal des nœuds car le scénario de déploiement est prédéfini [32]. Cependant, elle nécessite la connaissance des informations a priori sur la région d'intérêt qui ne sont pas toujours disponibles dans de nombreux environnements d'application, tels que les zones de guerre et les zones imprégnées de gaz toxiques.

2.3.2 Déploiement aléatoire

Dans le déploiement aléatoire les capteurs sont dispersés dans la zone de surveillance, typiquement sans planification ni aucune connaissance de la région. Ce déploiement convient aux régions où la présence humaine est difficile (par exemple, les zones sinistrées, les champs de bataille, la pollution atmosphérique et les feux de forêt). Ce type de déploiement est utilisé dans de nombreuses applications RCSF en raison de la simplicité de la distribution des capteurs [10]. Cependant, le problème de ce déploiement est qu'il n'est pas certain que chaque point de la zone de déploiement puisse être couvert par au moins un capteur (trous de couverture), et il est inévitable que certaines parties soient peu couvertes et d'autres densément couvertes (capteurs redondants). Par conséquent, une phase de post-déploiement est nécessaire pour relocaliser les capteurs redondants afin de couvrir les trous de couverture. Deux approches existent dans la littérature afin d'assurer le bon emplacement des capteurs :

- **L'auto déploiement** : consiste à déployer des capteurs mobiles aléatoirement et de les laisser se déplacer vers les emplacements désignés grâce à la communication et à la coordination entre les capteurs [25].
- **Déploiement aléatoire basé transport** : consiste à utiliser un ou plusieurs robots mobiles comme véhicule pour transporter les capteurs statiques déployés de manière aléatoire et les déposer aux endroits appropriés.

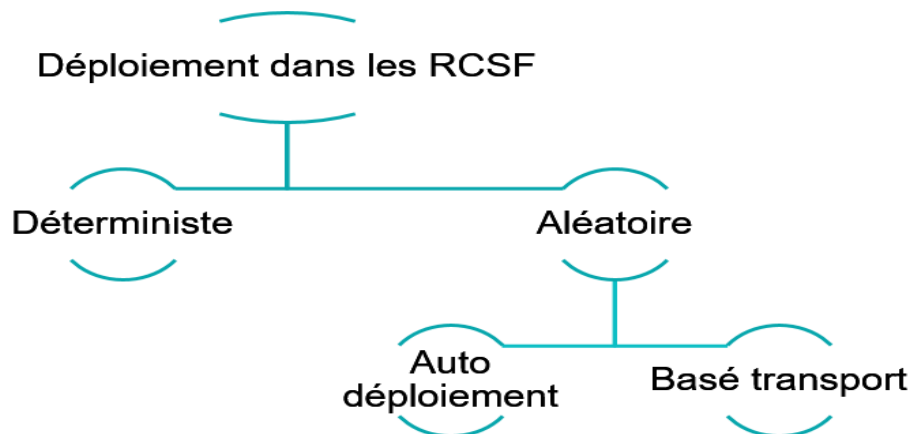


Figure 2.3: Classification des solutions de déploiement des capteurs

2.4 Solutions Antérieurs

2.4.1 Solution déterministe

2.4.1.1 Approche basée sur l'optimisation par colonies de fourmis (ACO-Greedy)

Liu et He [33] ont étudié les défis de couverture du RCSF basé sur une grille avec une connectivité garantie et un faible coût (GCLC), et ils ont proposé une solution de déploiement ACO-Greedy pour résoudre ce problème. Cette méthode améliore l'optimisation de la colonie de fourmis (ACO) et ajoute une nouvelle fonctionnalité, la migration avide des fourmis. ACO est un algorithme intelligent bien connu dans lequel des comportements combinés complexes sont générés à partir du comportement des fourmis. Dans le cadre de la plupart des algorithmes d'intelligence artificielle en plein essor, il est extrêmement efficace pour résoudre les problèmes d'optimisation combinatoire NP-difficiles. Dans la méthode basée sur ACO-Greedy, le champ de détection est composé de points de grille discrets, et des capteurs peuvent être installés sur ces points de grille et identifier les points d'intérêt (PoI) dans le rayon de détection. Le but du problème GCLC est de trouver un groupe de points aussi petit que possible parmi les points de grille candidats, et donc de déployer un nœud sur chaque point de grille individuel du groupe. En fin de compte, tous les points d'intérêt peuvent être couverts par des nœuds déployés. Chaque membre de cet groupe est appelé un point de solution (PoS). De plus, chaque PoS individuel doit être connecté au puits via un ou plusieurs sauts. Les puits sont des nœuds qui rassemblent les données des nœuds sans fil voisins et les transmettent à la station de base. Par exemple, **Figure 2.4** donne une solution au problème GCLC, où le point noir représente PoI et le point rouge représente PoS. Dans ce cas, les étoiles vertes représentent des puits, et en plus, l'ombrage symbolise la portée de détection/communication du nœud

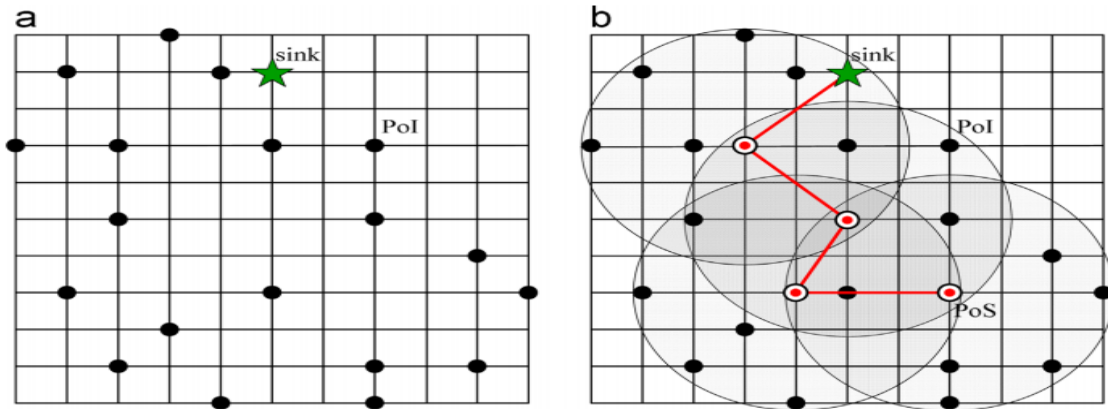


Figure 2.4: Un exemple du problème GCLC. (a) Avant le déploiement du nœud et (b) Après le déploiement du nœud)[33]

Discussion :

Cette stratégie permet de compléter rapidement une couverture complète et de réduire considérablement les coûts de déploiement. De plus, ACO-Greedy peut ajuster dynamiquement le rayon de détection/communication pour atténuer le problème de trou d'énergie et équilibrer la consommation d'énergie entre les nœuds de capteurs, prolongeant ainsi la durée de vie du réseau dans le réseau basé sur RCSF. Cependant, en ajoutant plus de nœuds dans les zones à fort trafic, le coût augmentera.

2.4.2 L'auto déploiement

2.4.2.1 Approche basée sur l'optimisation par essaims de vers lumineux (GSO)

Une structure de déploiement de capteur est introduite par Liao et al [34] qui s'appuie sur Glowworm Swarm Optimization (GSO) afin d'assurer la couverture après le déploiement initial aléatoire du capteur. L'intelligence en essaim est une sorte d'intelligence artificielle basée sur le mode de coopération des systèmes de décentralisation et de contrôle automatique. Elle met l'accent sur l'exploration du comportement collectif, qui est préparé par un groupe d'agents non requis interagissant localement les uns avec les autres et avec leur environnement. Bien qu'il n'y ait pas de contrôle centralisé pour déterminer le comportement de l'agent, l'agent suit quelques règles simples pour interagir localement avec d'autres agents, ce qui déclenche généralement une tendance générale.

Dans cette stratégie, chaque nœud capteur est considéré comme une luciole unique, qui libère une substance lumineuse appelée fluorescéine. La quantité de fluorescéine est déterminée par la distance entre le nœud capteur et ses capteurs voisins. Le nœud capteur est attiré vers la direction de son voisin avec une intensité de fluorescéine plus faible et choisit de se déplacer vers l'un d'eux. En utilisant cette méthode, la couverture du champ de détection est maximisée lorsque le nœud de capteur a tendance à se déplacer vers la zone avec la densité de capteurs la plus faible. Dans la **Figure 2.5** il y a trois capteurs : les capteurs 1, 2 et 3. Nous supposons que le capteur 3 a une luminance plus faible que les autres. Tel que (a) représente l'état initiale, (b) Le mouvement de capteur 1 vers 1' , (c) Le capteur 2 vers 2' et (d) représente le mouvement des capteurs 1 et 2 avant et après.

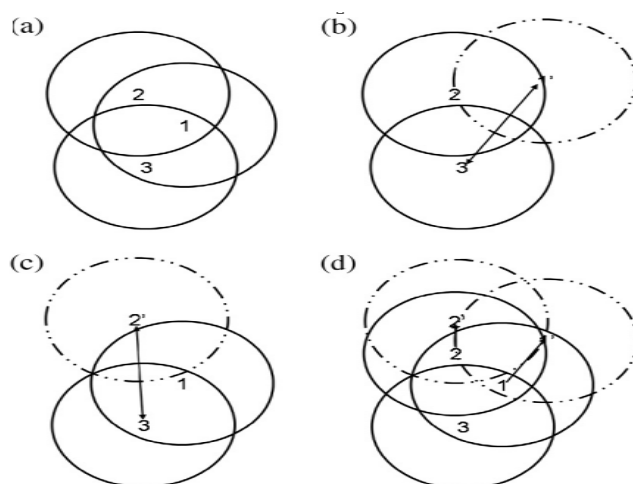


Figure 2.5: Illustration du mouvement du capteur[34]

Discussion:

Cette méthode de déploiement s'adapte facilement aux grands réseaux de capteurs, car aucun contrôle centralisé ne dicte le comportement des agents. Les résultats des simulations ont montré que l'approche de déploiement des capteurs basée sur le GSO peut fournir une couverture élevée avec un mouvement limité des nœuds de capteurs, ce qui peut être considéré comme les avantages de cette stratégie.

2.4.2.2 Algorithme de déploiement de capteurs assisté par des mouvements localisés pour la détection et la réparation de trous (HEAL)

Senouci et al [35] ont proposé une méthode pour résoudre le problème de détection et de réparation des trous dans les RCSF. Ils ont proposé une solution complète, distribuée et partielle de réparation basée sur des forces virtuelles pour la détection et la réparation de trous, appelée HEAL (hole detection and repair).

HEAL est un algorithme distribué et localisé qui s'exécute en deux étapes différentes. L'étape initiale comprend trois sous-tâches : la reconnaissance des trous, la découverte des trous et la détection des limites. Les auteurs ont proposé un algorithme de détection de trous distribués et locaux (DHD) qui opère sur le graphe de Gabriel du réseau (un graphe de Gabriel est un graphe qui relie un ensemble de points dans le plan euclidien). Bien que la distribution et la densité des nœuds soient différentes, DHD est très simple et peut gérer des trous de différentes formes et tailles. L'étape suivante utilise un nouveau concept de « zone de réparation des trous » pour traiter la réparation des trous. Il se compose de deux sous-tâches : déterminer la zone de cicatrization du trou et déplacer les nœuds. Les chercheurs ont proposé une méthode de guérison locale basée sur des

forces virtuelles distribuées, basée sur la Hole Healing Area (HHA), où les forces seront efficaces. Cela permet une guérison locale, où seuls les nœuds à une distance appropriée du trou participeront au processus de guérison. Le gestionnaire de trou (HM) sera chargé de déclarer que le trou a cicatrisé. Après avoir déterminé le HHA, HM informe les nœuds impliqués dans le processus de guérison. Le nœud recevant la force du centre du trou se déplace vers lui. Les chercheurs tentent d'équilibrer le compromis entre l'attractivité et la force répulsive pour restaurer les pores sans effets secondaires, comme le montre la (**Figure 2.5**). Le centre du trou applique une force d'attraction sur chaque nœud de la HHA situé à une certaine distance. En même temps, pour minimiser la couverture superposée, une force répulsive s'exerce entre les nœuds situés à une certaine distance.

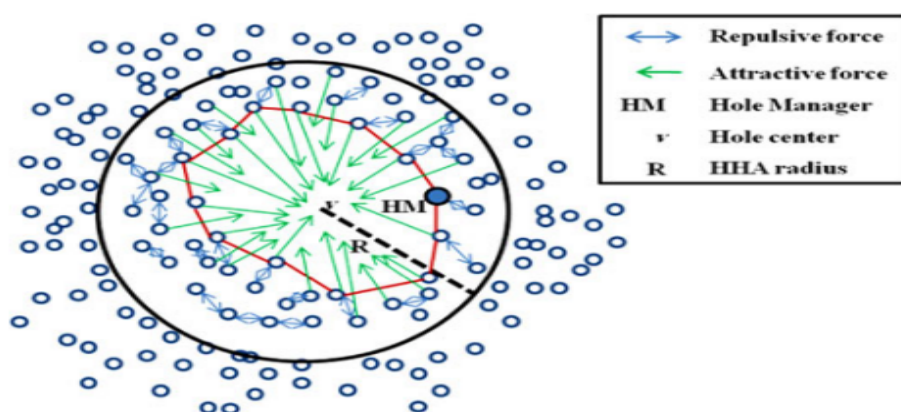


Figure 2.6: Principe du processus de guérison [35]

Discussion:

Les résultats expérimentaux ont montré que HEAL fournit une solution rentable et précise pour la détection et la réparation des trous dans les réseaux mobiles sans fil.

2.4.3 Déploiement aléatoire basé Transport

2.4.3.1 Based Sensor relocation (CBR)

Les auteurs [36] proposent une série de solutions de localisation : R3S2, G-R3S2 et CR3S2. Les capteurs sont déployés de manière aléatoire dans la ROI dans ces algorithmes. Afin de chercher et collecter les capteurs redondants, des robots sont utilisés pour les placer dans les trous de couverture trouvés par ces robots.

a) Algorithme R3S2

Une fois l'étape de découverte du voisinage terminée, chaque capteur détecte s'il existe des arcs dans son rayon de couverture qui ne sont pas couverts par ses voisins actifs [36]. Le robot se déplace de manière aléatoire, et communique avec le capteur entre deux déplacements consécutifs via un message "beacon" avec sa position actuelle. Après avoir reçu ce message, les capteurs actifs voisins répondent en indiquant la position et les trous de couverture des capteurs redondants adjacents. À la fin de cette étape, le robot prendra la décision suivante (remplir un trou, ramasser le capteur redondant, les deux en même temps, marche aléatoire)

Discussion : Perte d'énergie (zones visitées plusieurs fois par plusieurs robots).

b) Algorithme Grid-Based R3S2

La version à grille (G-R3S2) [37] impose certaines contraintes sur le mouvement du robot pour réduire l'aléatoire (le mouvement du robot est limité par la grille virtuelle), raccourcir le temps de traversée attendu et réduire le temps de réparation. Pour cela, les robots tentent d'explorer les zones les moins visitées du réseau et augmentent les chances de trouver des capteurs redondants et des trous de couverture. Il est à noter que chaque robot possède un identifiant et un numéro de série, qui peuvent définir l'accès à un point de grille. Le capteur le plus proche du point de grille maintient le nombre de fois que le robot visite chaque direction à partir de ce point. Après avoir reçu le message du robot, le capteur actif lui envoie un message, qui contient la direction la moins récemment visitée : le principe de l'algorithme LRV. Un choix aléatoire est effectué uniquement en cas d'égalité ou en cas d'absence de capteurs actifs dans le rayon de communication des robots.

Discussion :

- Les mouvements des robots ne sont pas optimisés.
- Charge de robot (le nombre max de capteurs que le robot peut charger à la fois) égale à 1.

c) Algorithme Cluster-Based R3S2

La particularité de cette méthode est d'utiliser CDS (Dominant Connected Set) pour déterminer la priorité du réseau en créant un cluster. Le choix du tête de cluster consiste à sélectionner le nœud avec le plus grand ID. Ce dernier implique quatre grilles adjacentes et combine les informations sur les vulnérabilités de couverture et les capteurs redondants des membres du cluster en une seule information qui est propagée au sein du groupe afin que tous les membres puissent l'utiliser. Le robot communique alors avec un capteur qui lui fournit une vue globale du cluster auquel il appartient [37].

Discussion :

- l'opération de clusterisation est très Coûteuse.
- CH(tête de cluster) concerne une surface très limitée (4 grilles adjacentes).

2.4.3.2 Algorithme Carrier-Based Sensor Deployment Algorithm for Perception Layer in the IoT Architecture

Cet algorithme[28] considérons un ROI constitué d'un ensemble de capteurs statiques (S) qui sont déployés par la méthode de largage/hélicoptère et ont le même rayon sensoriel. La ROI est divisé en grilles. Chaque grille est indexée par un tuple $g(x, y)$, la **Figure 2.7** illustre le partitionnement du système de grille de RCSF.

$g(0,0)$	$g(0,1)$	$g(0,2)$	$g(0,3)$	$g(0,4)$
	+4	+2	-5	
$g(1,0)$	$g(1,1)$	$g(1,2)$	$g(1,3)$	$g(1,4)$
+2				
$g(2,0)$	$g(2,1)$	$g(2,2)$	$g(2,3)$	$g(2,4)$
		+4		
$g(3,0)$	$g(3,1)$	$g(3,2)$	$g(3,3)$	$g(3,4)$
	-3			-2
$g(4,0)$	$g(4,1)$	$g(4,2)$	$g(4,3)$	$g(4,4)$
			+2	

Figure 2.7: la partition d'un système de grille d'un RCSF [28]

Il y a un robot mobile dans la ROI qui connaît sa position via le GPS connecté et connaît toutes les informations de position du capteur tout en parcourant toute la zone le long du trajet de la tondeuse. Après avoir parcouru la ROI, le robot apprend l'état de couverture de chaque grille. Pour chaque grille contenant des zones non couvertes, l'algorithme [28] recherchera des capteurs redondants dans les grilles environnantes, en partant de la plus proche, jusqu'à ce que la valeur négative change. positif. Lors de cette recherche, la grille qui manque le plus de capteurs (c'est-à-dire la grille avec la plus grande surface non couverte) sera préférée.

Discussion :

- L'algorithme est efficace pour les petites zones, non pas pour les grandes zones.
- La charge du robot doit être supérieure ou égale au nombre de capteurs manquant le plus dans la grille

2.5 Tableau comparatif des algorithmes de déploiement

Le tableau suivant représente une comparaison des algorithmes de déploiement étudiés précédemment.

Algorithme	Type	Centralisé / Distribué	Capacité du robot	Marche du robot
ACO-Greedy	Déterministe	Centralisé	/	/
GSO	Auto- déploiement	Distribué	/	/
HEAL	Auto- déploiement	Distribué	/	/
R3S2	Basé Transport	Distribué	1	LRV
Grid-Based R3S2	Basé Transport	Distribué	1	LRV
Cluster-Based R3S2	Basé Transport	Distribué	N	Aléatoire
Carrier-Based Sensor Deployment Algorithm for Perception Layer in the IoT Architecture	Basé Transport	Distribué	1	Aléatoire

Table 2.1: Tableau comparatif des algorithmes de déploiement

2.6 Conclusion

Afin d'assurer une meilleure couverture du réseau, nous avons mentionné dans ce chapitre les différentes techniques de déploiements dans les réseaux de capteur sans fil, ainsi que les différentes solutions proposées.

Chapitre 3

Conception

3.1 Introduction

Dans RCSF, les capteurs peuvent être déployés de manière aléatoire ou déterministe en fonction de l'application et des informations régionales. Dans le déploiement déterministe, chaque capteur est placé à une coordonnée prédéterminée, ce qui n'est généralement pas possible sur les champs de bataille ou les zones dangereuses. Dans ce cas, le déploiement aléatoire devient la seule option. Ce type de déploiement ne nécessite aucune connaissance préalable du terrain et est plus facile à mettre en œuvre. Cependant, en cas de déploiement aléatoire, il est probable que la zone d'intérêt ne soit pas totalement couverte (trous de couverture), et qu'une partie de la zone puisse être densément couverte (capteurs redondants).

Afin de garantir une couverture complète de la zone dans le cas où les capteurs sont déployés de manière aléatoire, nous proposons dans ce chapitre une nouvelle solution de relocalisation pour déplacer les capteurs redondants à l'aide de robots mobiles, afin de réparer les trous de couverture. Pour assurer que le robot explorera avec succès (répare tous les trous de couverture) la totalité du ROI, nous utilisons une stratégie basée sur le principe de la réflexion de la lumière, car il a déjà été prouvé [10] que la lumière provenant de n'importe quelle pièce peut atteindre la totalité de la zone. De plus, pour améliorer le temps de relocalisation, le robot suit une marche aléatoire autour de l'axe généré par la réflexion de la lumière en introduisant la notion de probabilité.

3.2 Motivation

Typiquement, le déploiement aléatoire des capteurs est le plus couramment utilisé pour une large gamme d'applications dans des environnements inaccessibles en raison de la simplicité de la distribution des capteurs, de la nature non planifiée de ce type de déploiement un RCSF peut entraîner des trous de couverture qui diminuent considérablement la fiabilité du réseau. Afin de surmonter ce problème les nœuds de capteurs sont dispersés en haute densité. Bien que le déploiement dense minimise les trous de couverture permet la tolérance aux pannes et augmente la fiabilité des applications, il a ses propres inconvénients il maximise la redondance qui diminue l'efficacité énergétique La surveillance de la même région de la zone d'intérêt par de multiples capteurs implique un gaspillage d'énergie. Ce comportement est en contradiction avec la contrainte la plus critique d'un WSN économe en énergie. d'un WSN efficace en énergie, il est crucial de trouver une solution qui réduise la redondance afin d'assurer un bon taux de couverture et une bonne connectivité.

La plupart des solutions proposées à ce problème sont centralisées, ce qui inclut la collecte des emplacements de capteurs redondants et des trous de couverture via des processus précédents. Dans ce cas, la consommation d'énergie du nœud sera plus rapide, peu de solutions sont des algorithmes distribués où chaque nœud du réseau ne connaît que des informations locales et communique avec ses voisins; Utilisant la méthode distribuée LRV pour explorer une ROI. L'inconvénient avec cette méthode est l'existence de sous

régions non visitées et d'autres qui sont visités plusieurs fois. Ces solutions sont limitées en terme de charge de robot (charge=1) ce qui est insuffisant vu que dans le cas réel la charge du robot peut être supérieure à un. De plus, dans toutes ces solutions, les boucles engendrées par le robot (l'exploration d'une même sous zone à plusieurs itérations) même si la charge du robot est nulle et qu'il se retrouve ou se dirige vers un trou, or il ne pourra pas le réparer ce qui va augmenter le temps de relocalisation et le mouvement non optimal des robots.

Nos solutions sont basées sur un déploiement aléatoire, une charge limitée du robot supérieure à un et inférieure à N et sont basées sur le principe de réflexion pour la marche du robot afin de minimiser le temps de relocalisation.

3.3 Hypothèse

Initialement , un robot et un ensemble de nœuds capteurs $\{c_0 \dots c_n\}$ sont déployés de manière aléatoire dans la ROI et voici les hypothèses prises en considération dans cette recherche:

- Les capteurs ont le même rayon de sensation (R_s), ce qui limite leur zone de surveillance, et ont le même rayon de communication (R_c), qui correspond à la distance autorisée à communiquer avec ses capteurs voisins, où $R_c \geq 2R_s$.
- La couverture dont il est question dans ce mémoire est la couverture de chaque point de la ROI. Un point est couvert lorsqu'au moins un capteur couvre ce point.
- Un capteur peut détecter ou sentir tout événement dans son rayon de détection. La couverture est déterminée en fonction du rayon de sensation.
- Deux capteurs situés dans leur rayon de communication peuvent communiquer entre eux. Les voisins d'un capteur sont définis comme des nœuds de capteurs situés dans son rayon de communication.
- Le robots et les capteurs peuvent obtenir la connaissance de leur emplacement utilisent des appareils GPS ou d'autres technologies de positionnement efficaces.

3.4 structure de données

3.4.1 structure de données des capteurs

Dans la mémoire locale de chaque capteur se trouve les informations suivantes :

- Son identificateur (id).

- Sa position (x,y).
- Son rayon de sensation (R_s).
- Son rayon de communication (R_c).
- Son état : s'il est actif ou pas.
- La liste de ses voisins qui se trouvent dans sa zone de sensation.
- La liste de ses voisins qui se trouvent dans sa zone de communication.

3.4.1.1 Messages envoyés par le capteur

Le capteur a la possibilité de communiquer avec d'autres capteurs et robots à condition qu'ils soient dans sa zone de communication par des messages tel que :

- **Message de voisinage** : Chaque capteur va diffuser un message pour tous les capteurs qui se trouvent dans sa zone de communication pour la découverte de voisinage, le capteur qui reçoit ce message va considérer l'émetteur comme voisin direct.
- **Message de détection des capteurs redondants et trous de couvertures** : Chaque capteur qui reçoit une demande d'information du robot, doit envoyer la liste des capteurs redondants et trous de couverture qui sont dans leurs voisinage à ce robot.

3.4.2 structure de données du Robot

Dans la mémoire locale du robot se trouve les informations suivantes :

- Son identificateur (id).
- Sa position (x,y).
- Son rayon de communication (R_c).
- Nombre de messages échangées.
- Nombre de pas effectuées.
- Sa charge actuelle.
- Sa charge maximale.

3.4.2.1 Messages envoyés par le robot

- **Message de demande d'information :** Le robot diffuse un message de demande d'informations aux capteurs qui se trouvent dans sa zone de communication afin de collecter les informations à propos des capteurs redondants et trous de couvertures. Chaque capteur qui reçoit ce message vérifie dans son voisinage s'il y a des capteurs redondants ou des trous de couvertures et transmet ces informations au robot.
- **Message de mise à jour de la liste des capteurs redondants et de trous :** A chaque fois que le robot collecte des capteurs redondants ou répare des trous de couvertures, il envoie un message d'informations aux capteurs voisins pour mettre à jour leur liste des capteurs redondants ou la liste des trous de couvertures.

3.5 Technique de base

La solution prise est une solution décentralisée assistée par robot, ayant pour but de maximiser la couverture de la ROI. La réparation des trous de couverture causés par le déploiement aléatoire se fait en déplaçant les capteurs redondants vers les trous de couverture à l'aide d'un robot. Ce dernier, suit une trajectoire similaire à celle de la réflexion de la lumière. Ensuite, nous avons introduit dans notre solution la loi des grands nombre tout en assurant le principe de la réflexion de la lumière pour que le robot puisse réparer les trous de couverture de la ROI plus rapidement.

3.5.1 la réflexion de la lumière

La réflexion de la lumière [39] est le changement de direction que subit un rayon lumineux lorsqu'il traverse la surface séparant deux milieux d'indices de réfraction différents et qu'il reste dans le milieu initial. La loi de réflexion nous dit que si l'angle avec lequel un rayon lumineux frappe la surface d'un miroir (ou d'une surface polie) par rapport à la normale est déterminé, l'angle avec lequel ce rayon sera réfléchi de la surface de ce miroir sera le même. Soit un rayon lumineux incident arrivant sur une surface séparant deux milieux d'indices de réfraction différents avec un angle d'incidence i comme le montre la **Figure 3.1**. Il apparaît un rayon réfléchi définissant un angle de réflexion i' par rapport à la normale.

- 1ère loi de la réflexion : le rayon réfléchi se trouve dans le plan qui contient à la fois le rayon incident et la normale à la surface de séparation.
- 2ème loi de la réflexion : l'angle d'incidence i et l'angle de réflexion i' sont égaux : $i = i'$.

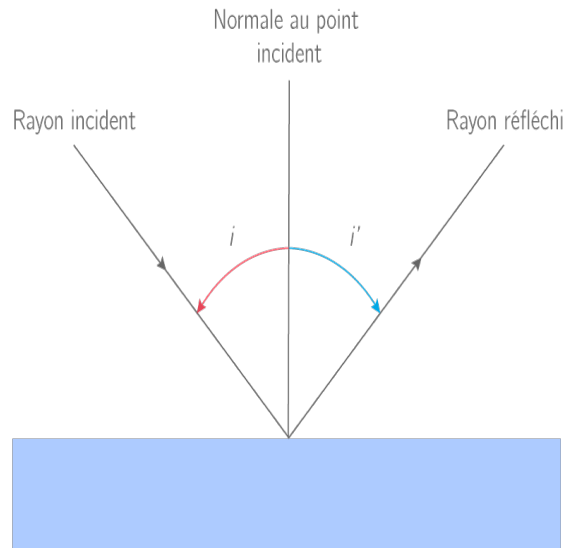


Figure 3.1: Principe de la réflexion de la lumière

Nous avons pris le choix d'utiliser le principe de réflexion dans notre solution car c'est déjà prouvé [10] que la lumière de n'importe quelle pièce peut atteindre toute la surface. Cela nous assure que si notre robot suit les mêmes principes, il explorera avec succès toute la ROI, détectera alors tous les trous et donc les réparera.

Lorsque le robot est à l'une des limites de la ROI, le robot applique le principe de réflexion : par rapport à la normale, il change de direction et sort au même angle d'arrivée comme le montre la **Figure 3.2**.

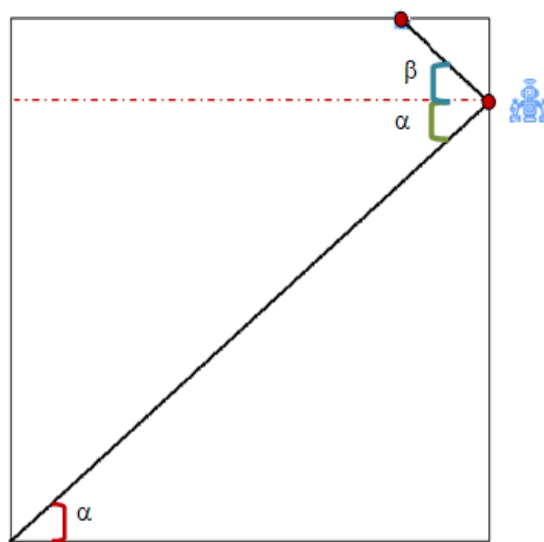


Figure 3.2: Exemple de l'utilisation du principe de réflexion

Nous avons remarqué en analysant le principe de réflexion que la trajectoire du robot sur une ligne droite n'est pas optimale, car la distance entre les différentes lignes droites est trop faible donc nous aurons un chevauchement d'informations, par contre l'application de la marche aléatoire à notre solution permet au robot de découvrir les trous de couvertures et les capteurs redondants autour de l'axe généré par la loi de réflexion de la lumière comme le montre la **Figure 3.3**.

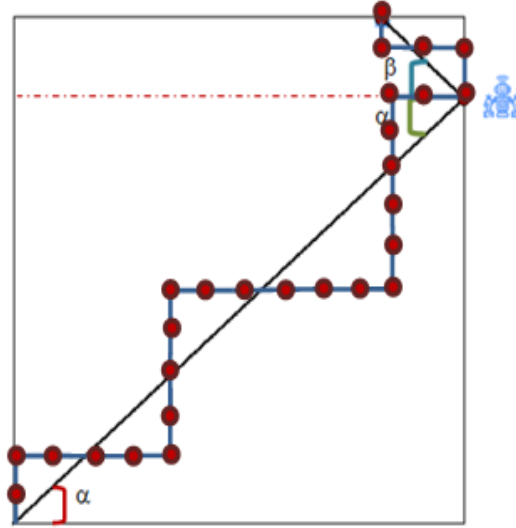


Figure 3.3: La marche aléatoire du robot

3.5.2 Loi des Grands Nombres

La loi des grand nombres est un théorème mathématique fondamental des probabilités et statistiques. Cette loi exprime le fait que les caractéristiques d'un échantillon aléatoire se rapprochent des caractéristiques statistiques de la population (ensemble d'individus ou d'éléments) lorsque la taille de l'échantillon augmente à l'infini. En d'autres termes, cela garantit que, lorsque le nombre de tirages effectués selon une loi de probabilité (comme les tirages successifs d'une pièce sur le côté pile ou face) tend vers l'infini, la moyenne empirique (moyenne calculée à partir des observations) converge vers la moyenne réelle d'une variable aléatoire suivant cette loi [40].

3.5.3 Relation entre la loi des grand nombre et la réflexion de la lumière

Grâce à la loi des grands nombres nous avons pu adopter la loi de la réflexion ; la **Figure 3.4** est un exemple illustratif sur la relation entre les deux.

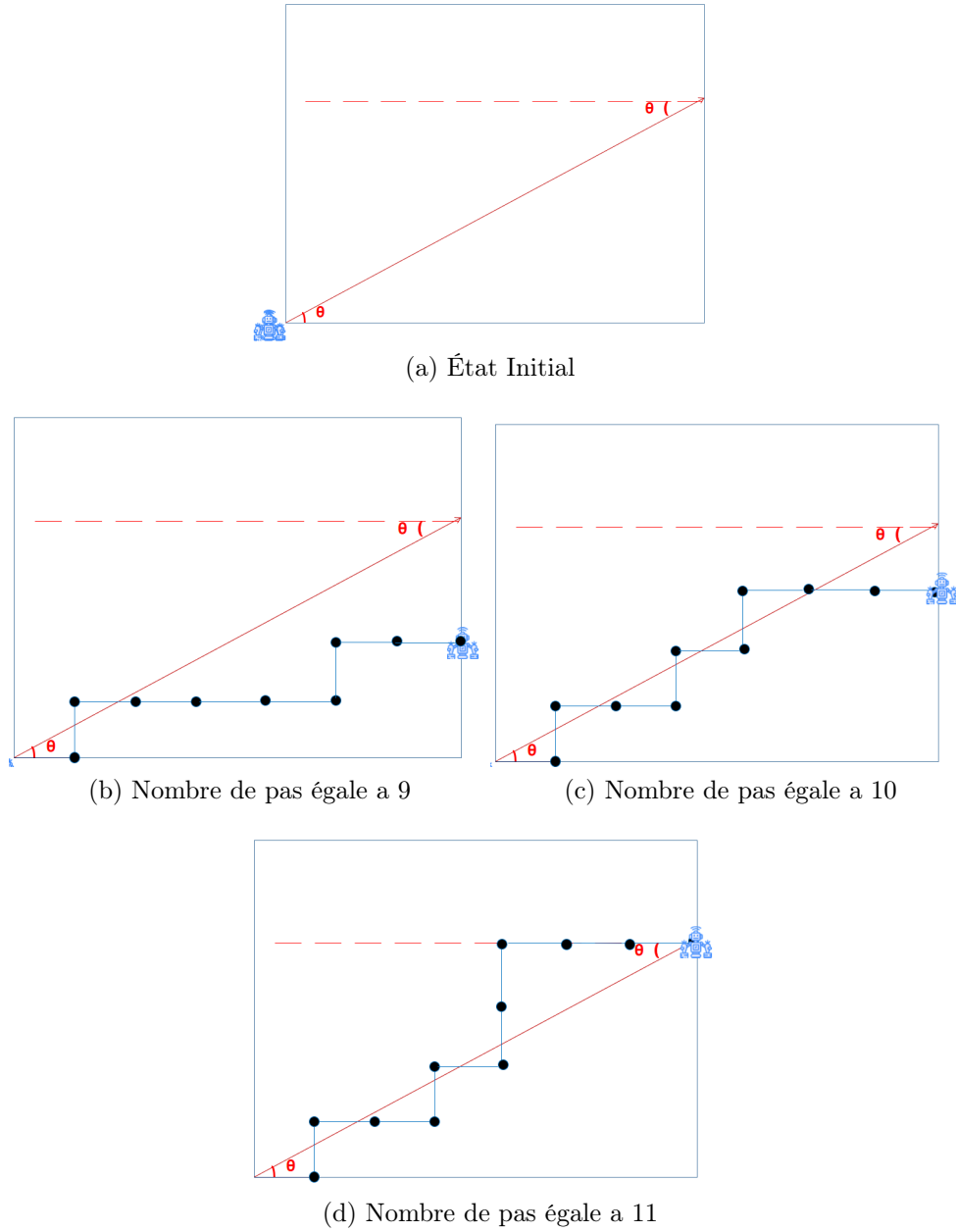


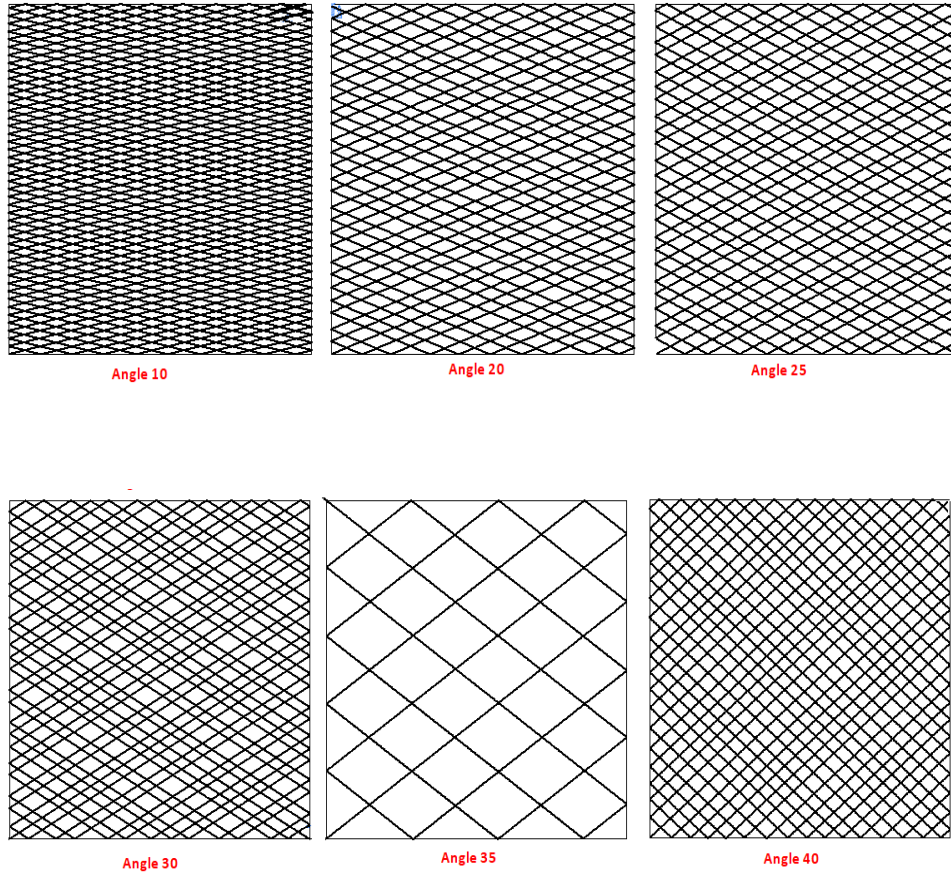
Figure 3.4: l'impact du nombre de pas du robot sur le principe de réflexion

Discussion:

Nous remarquons que lorsque le nombre de pas augmente, nous atteignons le principe de la réflexion, ce qui explique l'utilisation de grandes surfaces dans nos simulations afin d'atteindre le nombre nécessaire de pas du robot.

3.5.4 Angle de départ

En utilisant le principe de réflexion, nous devons démarrer avec un angle de départ (θ), et pour être sûr qu'il soit l'angle parfait nous avons fait des tests de découpage de surface de travail avec des différents angles comme montre les figures ci-dessous :



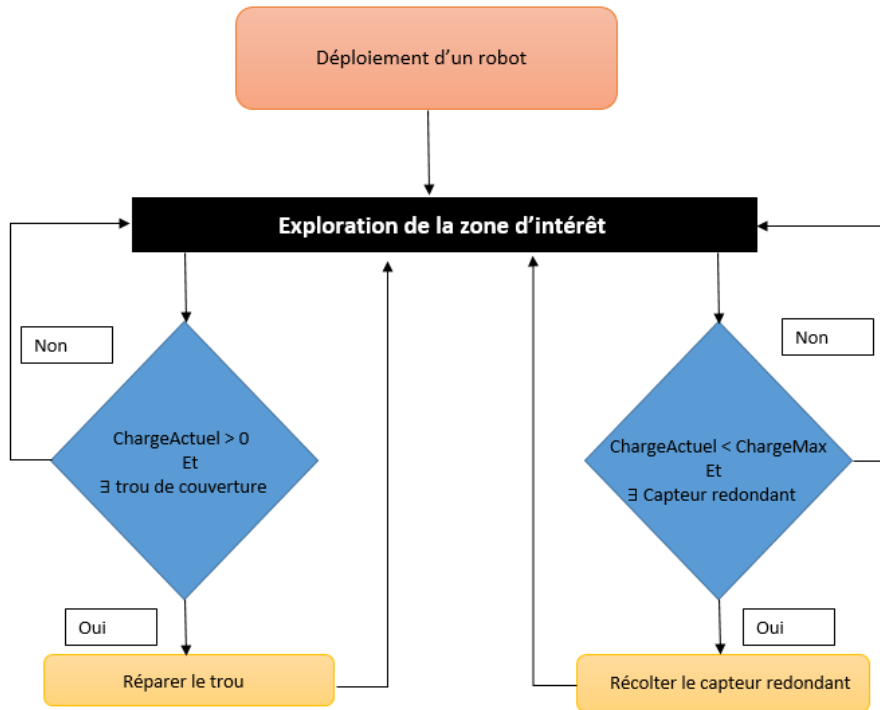
Discussion:

Chaque angle divise différemment la zone d'intérêt, mais pour la division dans notre cas, il est nécessaire d'utiliser des angles où la distance entre les axes de réflexion est plus grande que les petits, car plus la distance entre les différentes lignes est grande, moins il y aura d'informations superposées, donc l'exploitation de la zone se fait en un temps plus court.

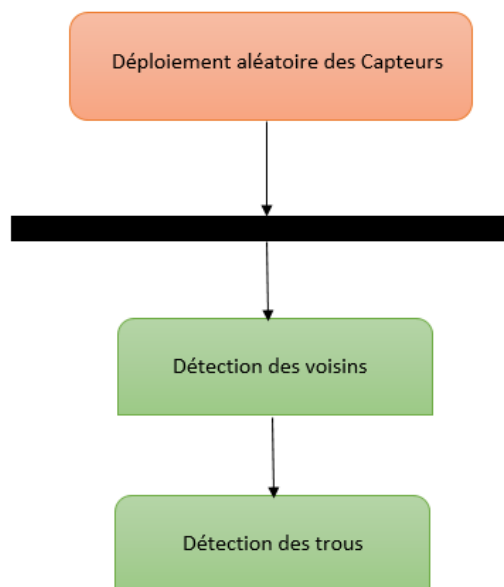
3.6 Algorithme proposé

Après le déploiement aléatoire de l'ensemble de capteur sur la ROI, l'étape de découverte de voisinage commence où chaque capteur envoie un message qui sera reçu par ses voisins.

Chaque capteur qui reçoit ce message considère l'émetteur comme un voisin direct. A partir des informations du voisinage, chaque capteur vérifie s'il est redondant ou a des trous dans son voisinage. Les organigrammes suivants montrent les principales fonctions réalisées par le robot et les capteurs.



(a) les principales fonctions réalisées par le robot



(b) les principales fonctions réalisées par les capteurs

3.7 Principe de Fonctionnement

3.7.1 Algorithme des Capteurs

Après le déploiement aléatoire, chaque capteur :

- Détecte son voisinage.
- Vérifie s'il est redondant.
- Vérifie s'il a des trous de couverture dans son voisinage.

3.7.1.1 Détection de voisinage

Chaque capteur diffusera un message à tous les capteurs situés dans sa zone de communication. Après réception de ce message, le capteur ajoutera ce capteur à sa liste de voisins.

Algorithme 1 : Algorithme de la détection de voisinage

Initialisation :

$Voisins_i = []$

À la réception d'un message Hello à partir d'un capteur j

Ajouter j dans l'ensemble de voisin $_i$

$Voisins_i = Voisins_i + j$

3.7.1.2 Détection capteurs redondant

Chaque capteur vérifie s'il est redondant. Un capteur est dit redondant :

- Si deux ou plusieurs capteurs sont superposés à la même position. Le capteur avec l'ID le plus petit sera actif et tous les autres seront redondants.
- S'il possède au moins deux voisins et tous les points d'intersections avec ses voisins (On prend en considération juste les points d'intersections qui sont dans le rayon R_s) sont couverts au moins par un quatrième capteur actif.

Algorithme 2 : Algorithme de la détection de capteurs redondants

Initialisation :Redondant $i = \text{True}$ Voisins $i = \text{ensemble des voisins de } i$ /*récupérer tous les points d'intersections entre voisins et vérifier s'ils sont
couvert par au moins 3 capteurs actifs */**pour chaque** *voisin* v **faire****pour chaque** *voisin* j **faire****si** $v \neq j$ **alors**/*récupérer les 2 points d'intersections entre i et j */Intersections = `getintersections (v.X, v.Y, j.X, j.Y)`

/*intersections = Xpoint1, Ypoint1, Xpoint2, Ypoint2

point1, point2 = Les coordonnées du 1^{er}, 2^{me} point d'intersection *//*Le nombre de capteurs actif qui couvre le 1^{er}/ 2^{me} point
d'intersection */

compteur1 = 0

compteur2 = 0

fin**pour chaque** *voisin* k **faire**/* A chaque fois qu'on trouve un voisin actif qui couvre ce point
d'intersection on incrémente le compteur*/**si** *point1 est couvert par i et k* **alors**

| compteur1 = compteur1 + 1

fin**si** *point2 est couvert par i et k* **alors**

| compteur2 = compteur2 + 1

fin**si** *compteur1 ≥ 3 and compteur2 ≥ 3* **alors**

| break.

fin**fin****si** *point1 est couvert par i and compteur1 < 3 or point2 est couvert par i
and compteur2 < 3* **alors**/*si un des points d'intersection est dans le rayon de i et il n'est pas
couvert par au moins 3 voisins actifs le capteur i n'est pas
redondant*/ Redondant $i = \text{False}$ break.**fin****fin****si** *non Redondant i* **alors**

| break.

fin**fin****si** *Redondant i* **alors**| envoyer le message "je suis redondant" à Voisins i **fin**

```

À la réception du message "je suis redondant" de j
/*Supprimer j de l'ensemble des voisins actifs de i*/
Voisini = Voisini - j

```

3.7.1.3 Détection de trou de couverture

Chaque capteur vérifie s'il y a des trous autour de lui, pour cela il vérifie si son intersection avec ses voisins est couverte par au moins un autre capteur actif, sinon il détecte un trou de couverture.

Algorithme 3 : Algorithme de la détection de trous de couverture

Initialisation :

```

trous i = []
Voisins i = ensemble des voisins de "i"
pour chaque j in Voisins i faire
    /*récupérer les 2 points d'intercections entre i et j dans la variable
    intersections*/
    intersections = getIntersections (i.X, i.Y, j.X, j.Y)
    /* intersections = Xpoint1, Ypoint1, Xpoint2, Ypoint2 */
    point1, point2 = Les coordonnées du 1er/2me point d'intersection
    /*Le nombre de capteur actif qui couvre le 1er et 2me point d'intersection*/
    compteur1 = 0
    compteur2 = 0
    pour chaque k in Voisins i faire
        si point1 est couvert par k alors
            | compteur1 = compteur1 + 1
        fin
        si point2 est couvert par k alors
            | compteur2 = compteur2 + 1
        fin
        si compteur1 < 2 and compteur2 < 2 alors
            | break.
        fin
    fin
    /*si point1 ou point2 n'est pas couvert par au moins 2 voisins ce point est
    considéré comme un début de trou */ si compteur1 < 2 alors
        | trous.append(point1) /* Ajouter point1 à la liste trous i */
    fin
    si compteur2 < 2 alors
        | trous.append(point2) /* Ajouter point2 à la liste trous i */
    fin
fin

```

3.7.2 Algorithme de Robot

Afin d'optimiser la trajectoire du robot et d'assurer l'ensemble du processus de la zone, nous sommes intéressées à la méthode de réflexion. Le robot a deux principales fonctions:

- l'exploitation de la zone afin de collecter les informations envoyées par les capteurs (capteurs redondants et trous).
- la réparation des trous de couverture.

3.7.2.1 Exploration de la zone par le robot

La marche du robot est basée sur le principe de la réflexion. Elle part d'un angle de départ et utilise la probabilité pas par pas pour choisir sa direction. Pour chaque pas (Δ), le robot tire une probabilité de valeur aléatoire selon deux valeurs P1 et P2.

Initialement le robot démarre avec un angle de départ θ , il tire une probabilité P pour choisir la direction en comparant P avec P1 et P2 :

- Si $P \leq P1$: avancer à droite / gauche, position actuelle $(x,y) \pm (\Delta,0)$.
- Sinon : avancer vers le haut / bas, position actuelle $(x,y) \pm (0,\Delta)$.

Cette étape se répète à chaque pas jusqu'à arriver à une des limites de la zone. en arrivant à une limite, le robot change sa direction selon le principe de réflexion et refait le calcul de probabilité. Le robot continue à explorer de la même façon jusqu'à la terminaison du processus de relocalisation.

Algorithme 4 : Algorithme de Marche robot

Initialisation : θ = Angle de départ $P1 = 1 / (1 + \tan \theta)$ $P2 = \tan \theta / (1 + \tan \theta)$ /*répéter à chaque pas du robot (Δ)*/**si** \exists des capteurs redondants dans mon voisinage **alors** **si** charge du robot < charge max **alors**

ramasser les capteurs redondants

charge du robot = charge du robot +1

fin**fin****si** charge du robot > 0 **alors**

réparer les trous s'il y en a

fin

/*on tire une probabilité aléatoire entre 0 et 1*/

P = Valeur aléatoire $\in [0,1]$ **si** $P \leq P1$ **alors**

/*on déplace vers la droite ou vers la gauche*/

 déplacer de (x,y) $\pm (\Delta,0)$.**fin**

jusqu'à (Fin du processus de relocalisation).

3.7.2.2 Réparation des trous

À chaque pas effectué, le robot écoute les messages transmis par les capteurs (capteurs redondants, positions de réparation):

- **Capteurs redondants** : S'il existe des capteurs redondants dans le voisinage et la charge maximale du robot n'est pas encore atteinte, ce dernier ramasse ces capteurs redondants jusqu'à ce qu'il atteigne sa charge maximale.
- **Positions de réparation** : S'il existe des trous de couverture et le robot chargé de capteurs redondants il les répare comme le montre la **Figure 3.1**.

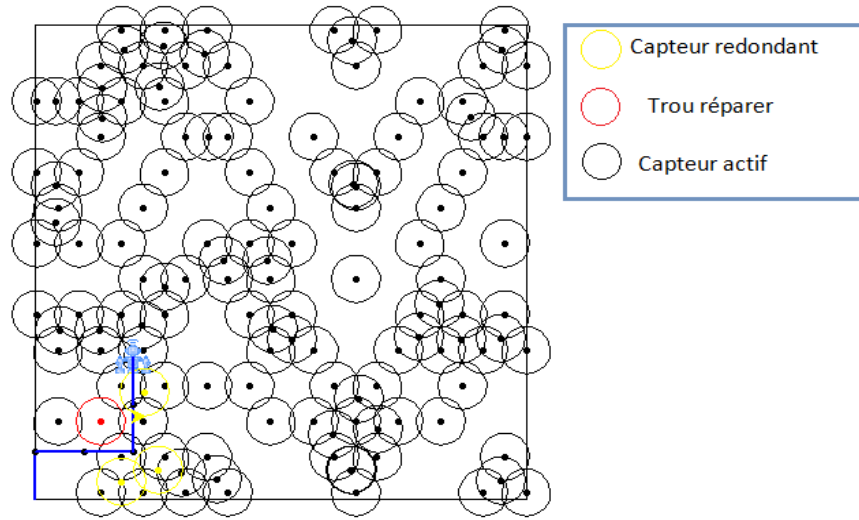


Figure 3.5: Exemple de réparation d'un trou

3.8 Version d'obstacle de notre solution

Notre algorithme tient également compte de l'existence d'obstacles qui ont un impact sur les réseaux de capteurs sans fil de trois manières différentes : communication, détection et mobilité. Dans cette sous-section, nous supposons que :

- Un capteur peut détecter l'existence d'un obstacle si et seulement si l'obstacle se trouve dans la zone de couverture.
- Les capteurs n'ont pas la capacité de couvrir les zones qui sont bloquées par des obstacles.

Arriver à un obstacle le robot change sa direction selon le principe de réflexion et refait le calcul de probabilité. le robot continue à explorer de la même façon jusqu'à la terminaison du processus de relocalisation.

la **Figure 3.5** représente un exemple de scénario en présence des obstacles, et comment le robot continue à explorer la ROI, jusqu'à ce que le processus de relocalisation se termine.

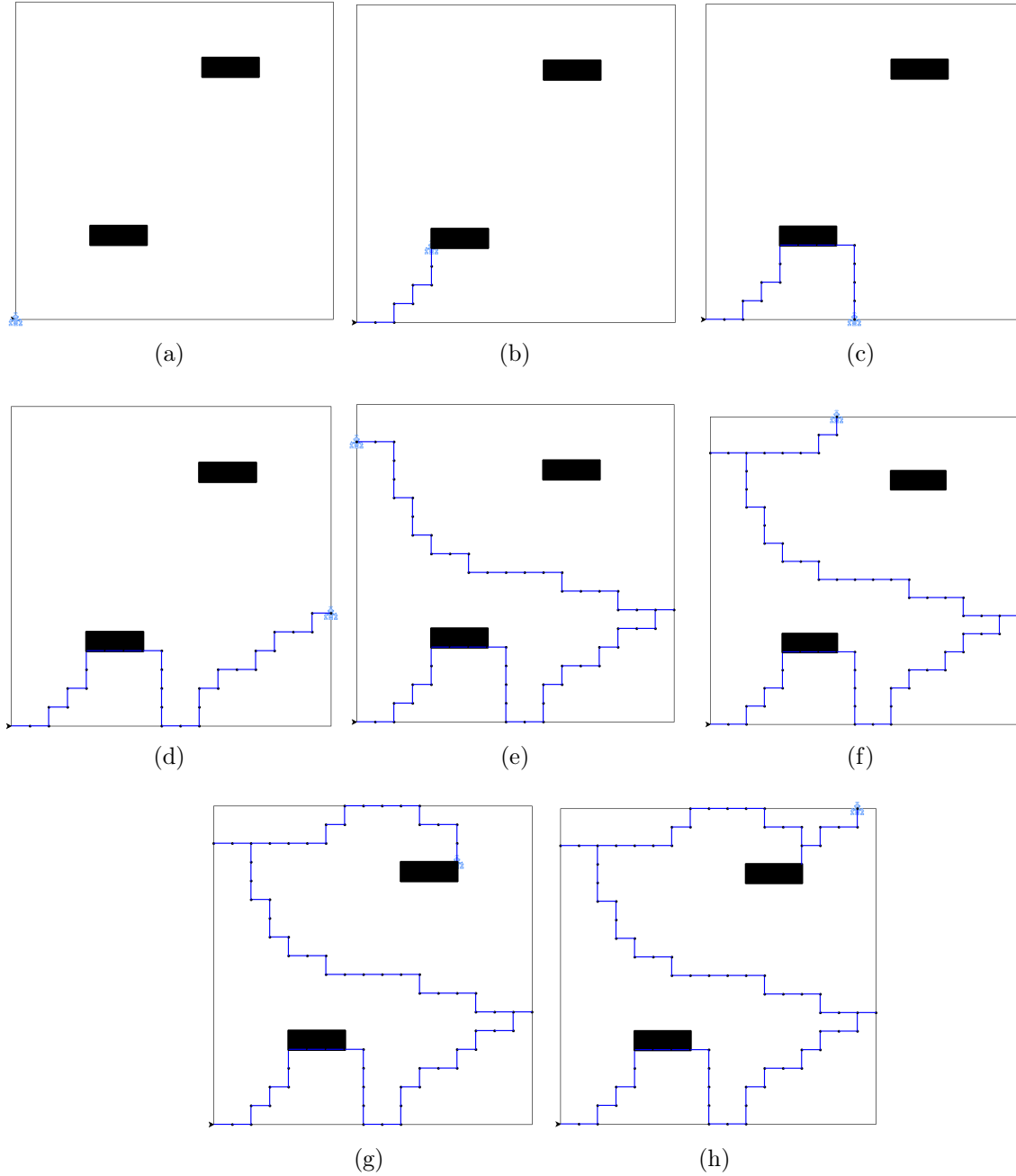


Figure 3.6: Exemple illustratif de la marche du robot en présence des obstacles

3.9 Conclusion

Dans ce chapitre, nous avons présenté les algorithmes de robot et de capteurs qui permettent l'augmentation de la couverture dans une ROI.

Dans notre solution, la marche du robot est basée sur la méthode de la réflexion qui permet d'explorer toute la surface du ROI. A chaque pas effectué par le robot, il s'échange des

messages avec les capteurs voisins afin de collecter les informations à propos de capteurs redondants pour les ramasser et de trous de couvertures pour les réparer. Nous avons aussi expliqué comment notre solution s'adapte également à la présence d'un obstacle suivant le même principe de réflexion.

Dans le chapitre suivant, à travers des différents scénarios de simulation nous allons évaluer certains critères de performance de notre solution proposée, tel que le temps de réparation, le taux de couverture total, nombre de messages échangés final et le nombre de pas effectué par le robot.

Chapitre 4

Simulation et évaluation des performances.

4.1 Introduction

Dans ce chapitre, nous introduisons d'abord les paramètres de simulation, puis nous détaillons l'environnement de simulation. Ensuite, nous évaluons les performances de nos algorithmes en nous basant sur la loi de la réflexion de la lumière LR-SR (marche aléatoire sur l'axe généré par la loi de la réflexion) et R-SR (marche en ligne droite sur l'axe généré par la loi de la réflexion) en les comparant à carrier-based sensor deployment algorithm[28] et l'algorithme LRV[41].

4.2 Outils de développement

Afin d'évaluer les performances d'algorithme proposé, nous avons choisi d'utiliser du langage Python version 3.8 qui permet la manipulation de différentes fonctions, mettre en oeuvre des algorithmes et visualiser le fonctionnement sur une interface graphique en implémentant différents packages tel que :

- **Threading** : s'appuie sur les fonctionnalités de bas niveau pour rendre le travail avec les threads encore plus facile et plus pythonique. L'utilisation de threads permet à un programme d'exécuter simultanément plusieurs opérations dans le même espace du processus. Dans le réseau chaque noeud est considéré comme un Thread.
- **Random** : est un module Python regroupant plusieurs fonctions permettant de travailler avec des valeurs aléatoires. La distribution des nombres aléatoires est réalisée par le générateur de nombres pseudo-aléatoires Mersenne Twister, l'un des générateurs les plus testés et utilisés dans le monde informatique.
- **Turtle** : Le module turtle de Python permet de commander une tortue qui va tracer des segments. Il est bien évidemment inspiré du fameux langage Logo développé dans les années 60. Nous utilisons Turtle afin de générer les dessins et animations présents dans notre application.
- **Math** : Ce module fournit l'accès aux fonctions mathématiques définies par la norme C.
- **Temps** : Ce module est une des façons les plus simples de manipuler le temps dans un programme. Un temps en python est par défaut un nombre représentant des secondes.

4.3 Environnement de la simulation

Nous avons obtenu ces résultats présentés à partir d'un réseau composé de noeuds capteurs déployés aléatoirement sur une zone d'intérêt carrée de différentes dimensions de manière

à obtenir un taux de couverture initial de 65% comme le montre la **Figure 4.1** et qui sera couverte au fur et à mesure.

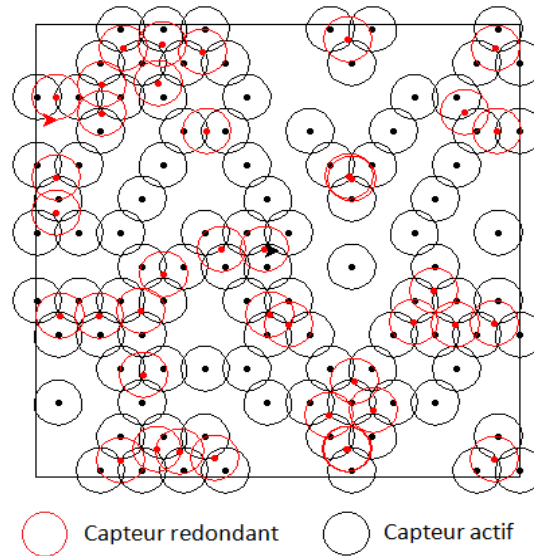


Figure 4.1: Exemple d'un environnement de simulation

4.4 Critère de performances

Lors de l'évaluation de l'algorithme, les critères de performances retenus sont :

- **Temps de relocalisation** : c'est le temps nécessaire pour la relocalisation des capteurs redondants dans les trous de couverture détectés afin de maximiser le taux de couverture.
- **Taux de couverture** : c'est le pourcentage de la zone couverte par au moins un capteur sur l'ensemble de la zone d'intérêt.
- **Nombre moyen de mouvements de robot** : c'est le nombre moyen de pas effectué par le robot pour l'exploration de la zone.

4.5 Paramètre de simulation

Afin de réaliser l'évaluation de notre solution "Light Reflexion based sensor relocalisation algorithm in wireless sensor and robot networks" (LR-SR), nous varions différents paramètres afin de distinguer et observer leur impact sur les critères de performances cités ci-dessus.

Paramètre	Explication
La surface de la ROI	représente la surface de la région d'intérêt
L'angle de départ θ	représente l'angle initial du robot qui définit l'axe générer par le principe de la loi de la réflexion de la lumière.
La charge du robot	c'est le nombre de capteurs redondant maximum que peut transporter le robot simultanément.
Taux de couverture	représente le taux initial après le déploiement des capteurs dans la ROI.
Pas du robot	représente le mouvement que fait un robot.

Table 4.1: Les paramètres de simulation

4.6 Résultats et évaluation des performances

Dans cette section, nous comparons les critères de performance de notre solution d'adaptation et d'autres solutions de déploiement en partant des mêmes paramètres énumérés dans les tableaux de simulation. Il est à noter que l'algorithme se termine lorsqu'on atteint un taux de couverture égale à 98%.

4.6.1 Impact de la variation de l'angle de départ sur le temps de relocalisation

Dans cette simulation, nous évaluons le temps de relocalisation avec un angle initial varié. Les paramètres de simulation détaillés de l'expérience sont listés dans le tableau ci-dessous:

Paramètres de simulation	
Surface de ROI	3240 * 3240
Pas du robot	2*Rs
charge du robot	8
Taux de couverture initial	65 %

Table 4.2: Simulation 1

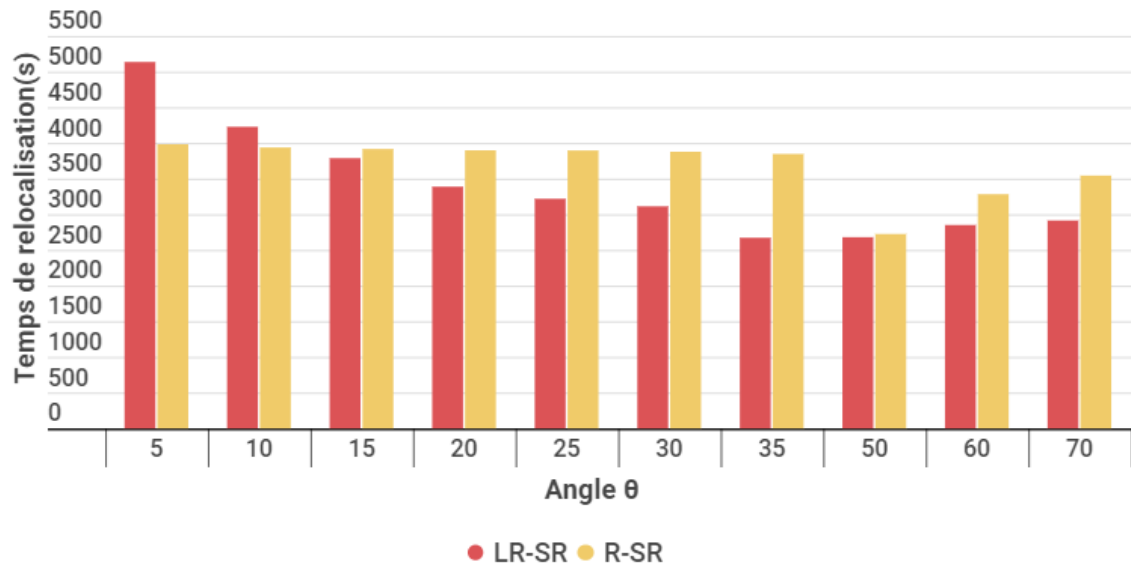


Figure 4.2: Impact de la variation de l'angle sur le temps de relocalisation

• Analyse

La **Figure 4.2** montre que pour atteindre un taux de couverture égal à 98%, l'approche LR-SR nécessite moins de temps par rapport à l'approche par réflexion (R-SR). Ceci est dû à la marche suivie par chacune des solutions, l'introduction de la marche aléatoire autour de l'axe dans la solution LR-SR a permis au robot d'explorer non seulement l'axe mais aussi ses environs en un seul passage ce qui réduit le temps total de relocalisation par rapport à celui qui suit une ligne droite.

Les meilleurs cas représentent les angles qui divisent la ROI de manière optimale, c'est-à-dire que le taux de chevauchement de l'exploration d'un axe avec le suivant est optimal, contrairement aux angles qui rapprochent les axes de réflexion et conduisent à une exploration lente, ce qui est dû au fait que le robot peut explorer presque les mêmes sous-zones de deux axes.

4.6.2 Impact de la variation de Pas du robot sur le temps de relocalisation

Dans cette simulation, nous évaluons le temps de relocalisation avec différents pas du robot. Les paramètres de simulation détaillés de l'expérience sont listés dans le tableau **Table 4.3**

Paramètres de simulation	
Surface de travail	3240 * 3240
Angle Initial	35°
charge du robot	8
Taux de couverture	65 %

Table 4.3: Simulation 2

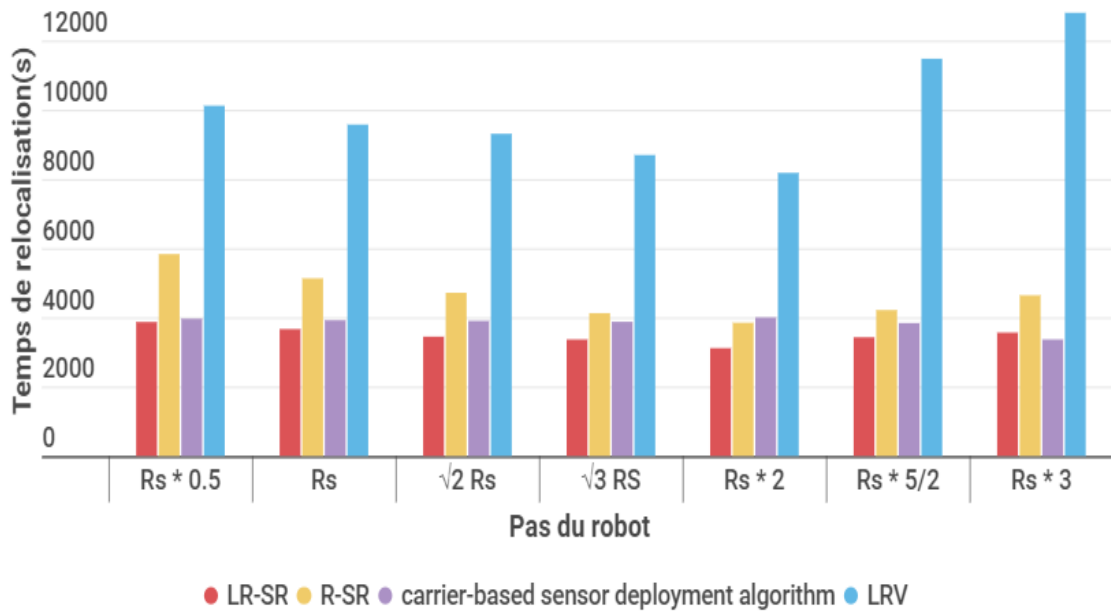


Figure 4.3: Impact de la variation de Pas du robot sur le temps de relocalisation

• Analyse

D'après la **Figure 4.3** nous remarquons que lorsque le pas du robot augmente, le temps de relocalisation nécessaire diminue ; cela est dû au fait que le robot s'arrête après chaque pas pour découvrir les trous et les capteurs redondants qui se trouvent dans son rayon de communication. Par conséquent, plus la taille du pas est petite en dessous du rayon de communication, plus les informations collectées sont redondantes, ce qui augmente le temps de relocalisation dans les algorithmes R-SR et LR-SR. Lorsque le pas du robot est égal à $2 \cdot R_s$, le robot découvre des informations d'une autre zone après chaque pas, ce qui donne le meilleur temps de relocalisation pour nos algorithmes. Dans le cas où la taille du pas est supérieure à $2 \cdot R_s$, le robot ne collecte pas les informations de toute la zone et la réparation prend plus de temps, ce qui augmente le temps de relocalisation. Alors que l'algorithme LRV est dû aux boucles de recherche, le robot peut rester longtemps à ré-explore une zone déjà

explorée au lieu d'aller dans les zones inexplorées et l'algorithme de carrier-based sensor deployment algorithm donne des temps irréguliers mais meilleurs que notre solution R-SR. Cela est dû au fait que dans l'algorithme R-SR, le robot peut ne pas atteindre certaines zones parce qu'il suit une ligne droite sur l'axe généré par la réflexion, contrairement à l'algorithme de carrier-based sensor deployment algorithm qui visite toutes les zones par le biais du balayage de la tondeuse, mais ce problème n'existe plus lorsque la marche aléatoire est adoptée.

4.6.3 Impact de la variation de la charge du robot sur le temps de relocalisation

Dans cette expérience, nous évaluons le temps de relocalisation en varions la charge du robot. Les paramètres de simulation détaillés de la simulation sont listés dans le tableau **Table 4.4**

Paramètres de simulation	
Surface de travail	3240 * 3240
Angle Initial	35°
Pas du robot	2*Rs
Taux de couverture	65 %

Table 4.4: Simulation 3

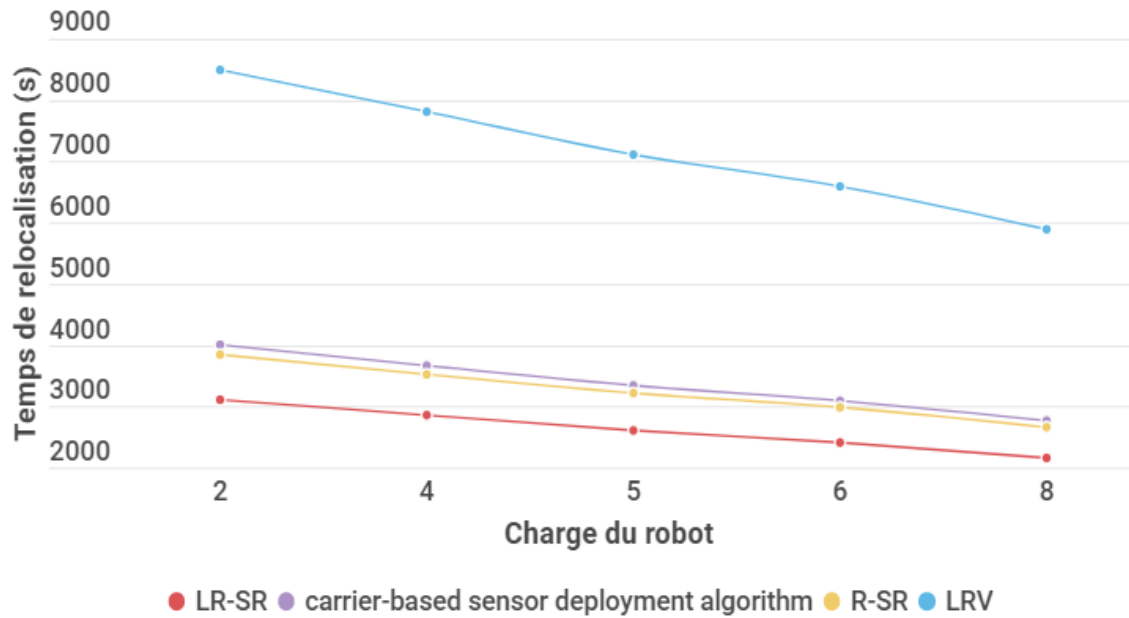


Figure 4.4: Impact de la variation de charge du robot sur le temps de relocalisation

• Analyse

Comme il est illustré dans la **Figure 4.3**, afin d'atteindre un taux de couverture égale à 98%, les résultats montrent que plus la charge du robot augmente, plus le temps de relocalisation diminue. L'augmentation de la charge du robot lui permet de ramasser plusieurs capteurs redondants, ce qui augmente sa chance d'être chargé par des capteurs lorsqu'il détecte un trou de couverture et donc il le répare directement par ces derniers au lieu de chercher des capteurs redondants.

4.6.4 Impact de la variation de la surface

• Sur le temps de relocalisation

Dans cette expérience, nous évaluons le temps de relocalisation avec différentes surfaces. Les paramètres de simulation détaillés de l'expérience sont listés dans le tableau :

Paramètres de simulation	
Pas du robot Δ	$2 * R_s$
Angle Initial	35°
charge du robot	8
Taux de couverture	$\approx 65\%$

Table 4.5: Simulation 4

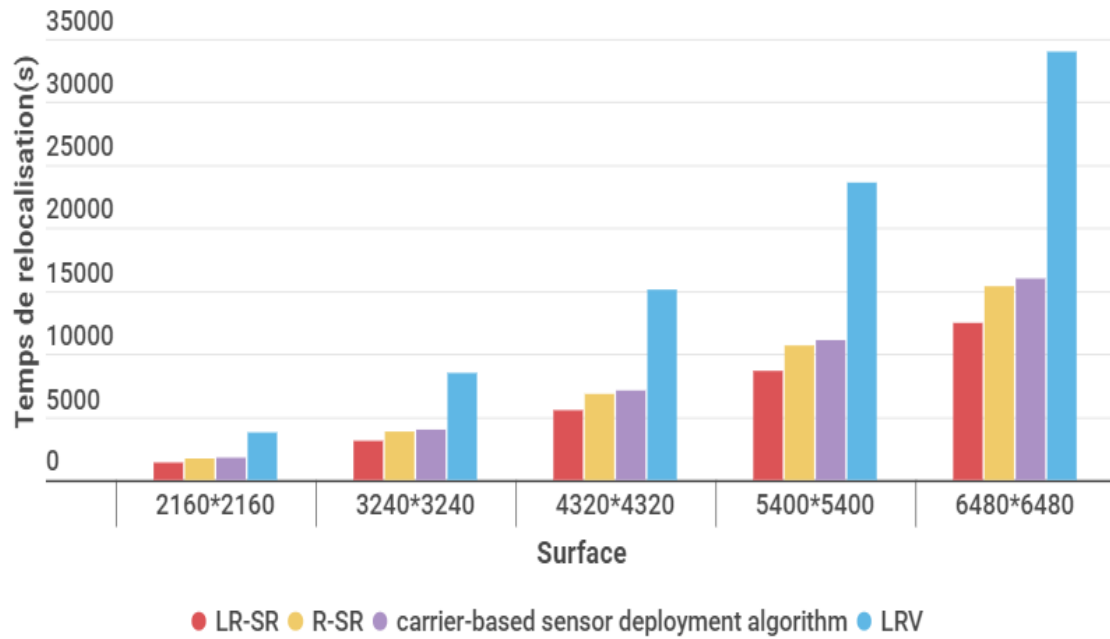


Figure 4.5: Impact de la variation de la surface sur le temps d'exécution

- Sur le nombre moyen de pas du robot

Dans cette expérience, nous évaluons le nombre moyen de pas du robot avec différent surface . Les paramètres de simulation détaillés de l'expérience sont listés dans le tableau **Table 4.5** .

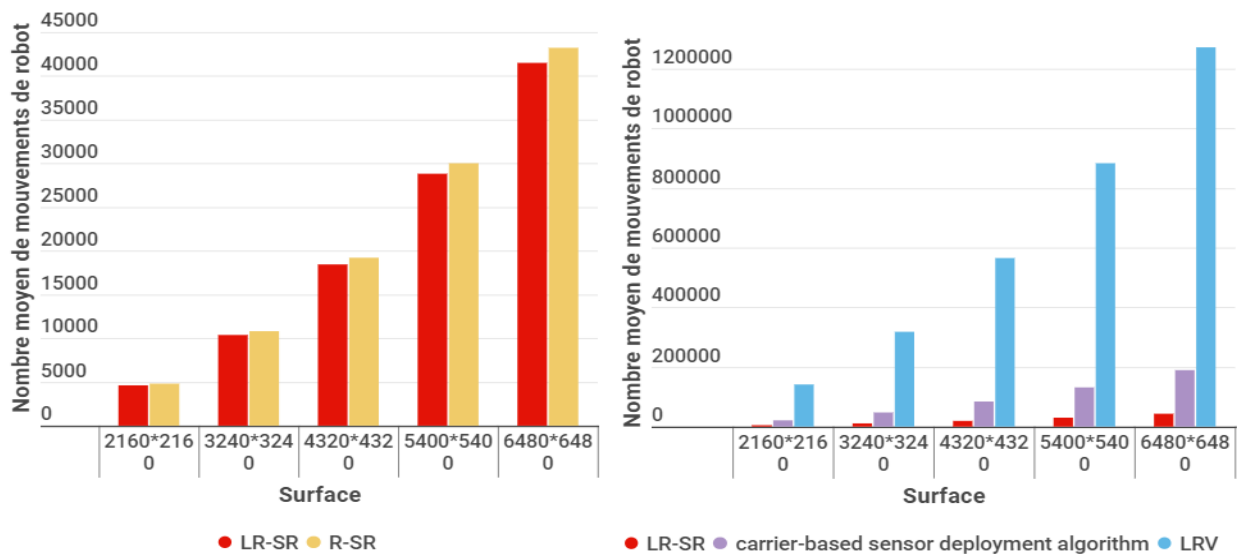


Figure 4.6: Impact de la variation de la surface sur le nombre moyen de pas du robot

- **Analyse**

A partir de ces Figure(**Figure 4.5** , **Figure 4.6**), nous pouvons clairement voir que plus la zone est grande, plus le temps de relocalisation et le nombre moyen de mouvements du robot augmentent et l'écart entre LRV, l'algorithme carrier-based deployment algorithme et l'approche LR-SR devient important, ce qui signifie qu'ils ne sont pas de bonnes solutions pour les grandes zones, cela s'explique par le fait que LRV st soumis à des boucles de recherche et que carrier-based deployment algorithme , pendant sa phase d'exploration, passe par des capteurs redondants sans les ramasser. Tandis que l'approche par réflexion(R-SR) qui donne des résultats proches a LR-SR et conserve une augmentation presque linéaire, mais Le problème avec la réflexion est que la trajectoire du robot sur une ligne droite n'est pas optimale, car la distance entre les différentes lignes droites est trop faible, donc l'application de la marche aléatoire à LR-SR permet en un seul passage de découvrir non seulement le chemin de l'axe généré par le principe de réflexion de la lumière mais aussi ces alentour. La solution LR-SR est donc plus adaptée aux grandes surfaces.

4.6.5 Comparaison de temps de relocalisation requis pour atteindre un certain taux de couverture

Nous étudions maintenant, le temps de relocalisation nécessaire pour atteindre un certain taux de couverture. Les paramètres de simulation détaillés de l'expérience sont listés dans le tableau :

Paramètres de simulation	
Pas du robot Δ	2 * Rs
Angle Initial	35°
charge du robot	8
Surface	3240*3240

Table 4.6: Simulation 5

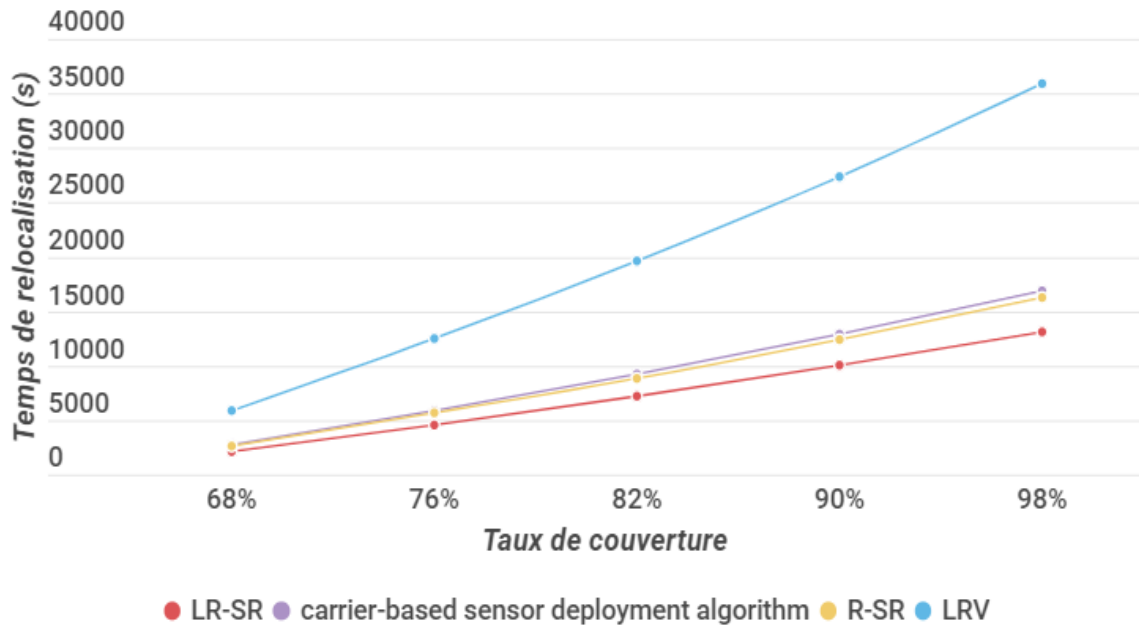


Figure 4.7: le temps de relocalisation requis pour atteindre un certain taux de couverture pour les différents solutions

• Analyse

Nous voyons dans la **Figure 4.7**, que le temps de relocalisation nécessaire à notre solution (LR-SR) pour atteindre n'importe quel taux de couverture est beaucoup plus petit que le temps nécessaire aux autres solutions pour atteindre le même taux. Cette différence est due au mode d'exploration utilisé, LRV nécessite un temps plus long car le robot dans LRV est soumis à des boucles de recherche, la réflexion est soumise à une marche droite non optimale du robot, même chose pour carrier-based sensor deployment algorithm qui souffre du temps perdu à découvrir l'ensemble de la zone avant la collecte des capteurs et/ou la réparation des trous, contrairement à notre approche dans laquelle le robot adopte une marche aléatoire autour de l'axe générée par le principe de réflexion de la lumière qui conduit à la découverte totale du ROI sans boucles et dans un temps optimal.

4.6.6 Impact de la présence d'obstacle sur notre solution

Dans cette simulation, nous évaluons le temps de relocalisation et le nombre moyen de mouvement avec la présence ou pas des obstacles dans notre solution. Les paramètres de simulation détaillés de l'expérience sont listés dans le tableau ci-dessous:

Paramètres de simulation	
Pas du robot Δ	$2 * R_s$
Angle Initial	35°
charge du robot	8
Taux de couverture	$\approx 65\%$

Table 4.7: Simulation 6

4.6.6.1 Sur le temps de localisation

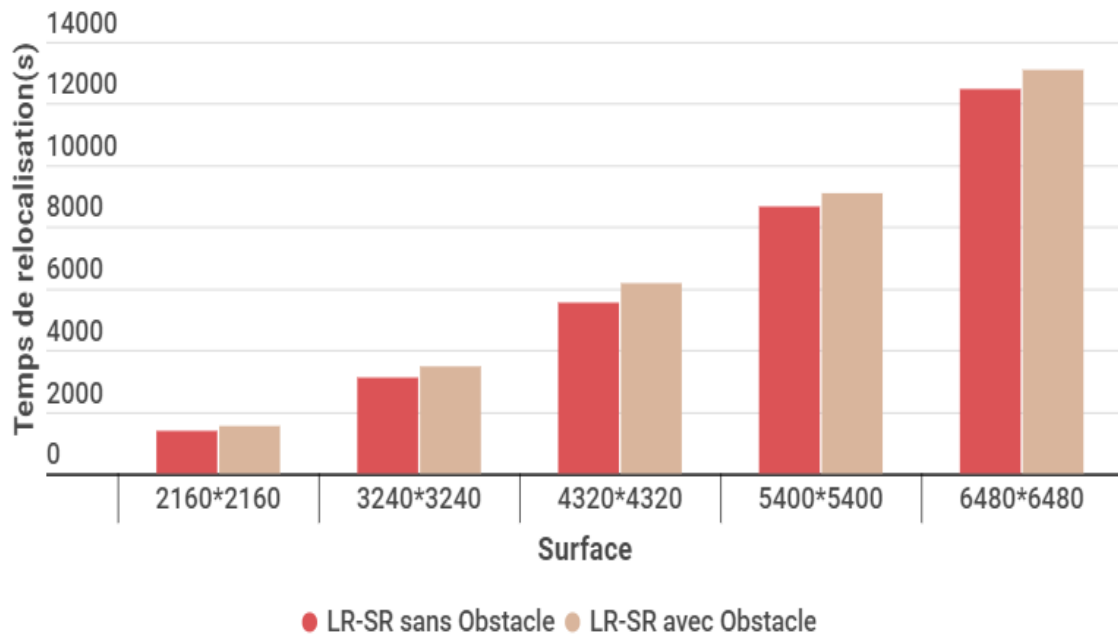


Figure 4.8: Comparaison du temps de relocalisation entre la présence ou pas d'un obstacle dans notre solution

- **Analyse :**

D'après la **figure 4.8** nous pouvons voir que la différence entre le temps de relocalisation nécessaire pour notre solution avec ou sans présence d'obstacles n'est pas très importante pour atteindre le même taux de couverture qui est égal à 98%. Bien que les obstacles aient une influence sur la marche du robot car à chaque fois que le robot rencontre un des obstacles, il appliquera le principe de réflexion de la lumière, mais notre solution a gardé la même performance et il est prouvé qu'elle est adaptable à ce type de région.

4.6.6.2 Sur le nombre moyen de mouvements de robot

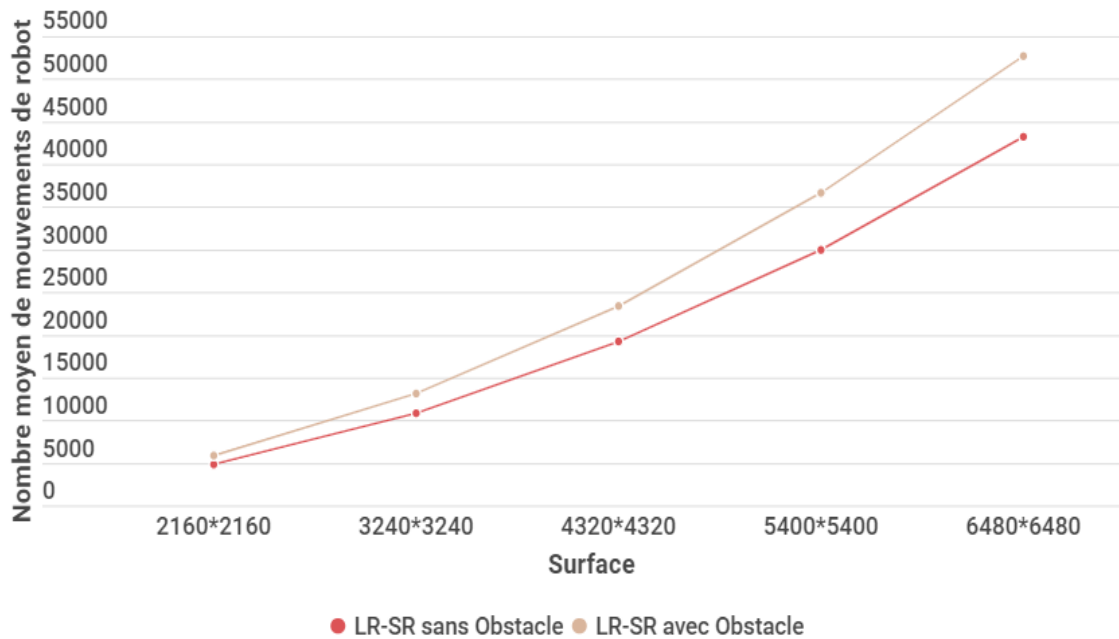


Figure 4.9: Comparaison du nombre moyen de mouvement du robot entre la présence ou pas d'un obstacle dans notre solution

- **Analyse :**

Nous analysons dans la **Figure 4.9** que la présence d'obstacles dans la surface augmente le nombre de pas du robot, et il est évident que cela est dû à l'application du principe de réflexion de la lumière chaque fois que le robot rencontre un des obstacles ou un des bords de la zone d'intérêt, mais la différence est très faible, voire négligeable, et on voit bien que notre solution est parfaitement adaptée à ce type d'application.

4.7 Analyse Générale

En analysant ces performances, les résultats de simulation ont montré que notre solution LR-SR qui vise à maximiser la couverture d'une région d'intérêt et à résoudre le problème de relocalisation des capteurs redondants dans les réseaux de capteurs et de robots sans fil en utilisant un principe de physique qui est le principe de réflexion de la lumière avec l'ajout de la notion de marche aléatoire en utilisant des formules mathématiques pour calculer les probabilités qui vont décider de la direction du déplacement du robot sur l'axe x permet de réduire le nombre moyen de mouvements du robot par conséquent réduire le temps de relocalisation et qu'il offre de meilleures

performances que LRV, R-SR et l'algorithme de déploiement basé sur le porteur en termes de temps de relocalisation et de nombre moyen de mouvements et d'optimisation de la trajectoire du robot. Par conséquent, notre approche répond aux objectifs énoncés précédemment.

4.8 Conclusion

Dans ce dernier chapitre, les performances de notre solution (LR-SR) et la solution LRV, la solution de carrier-based sensor deployment algorithm et la réflexion ont été analysées. Chaque solution a sa propre nature et donne des performances variables. Nous avons constaté que notre approche offre de meilleurs résultats avec ou sans la présence d'obstacle. De ce fait, on peut conclure que parmi les solutions étudiées, notre solution répond le mieux au besoin des réseaux de capteur sans fil.

Conclusion Générale

Il y a plus de dix ans, le paradigme de l'internet des objets (IdO) a été inventé, permettant aux ordinateurs d'accéder à des données sur les objets et l'environnement sans interaction humaine. Les RCSFs sont l'une des technologies traditionnellement considérées comme des catalyseurs clés du paradigme de l'IdO, apportant des capacités de détection et d'actionnement plus riches aux applications IdO. L'utilisation des réseaux de capteurs sans fil (RCSF) dans différentes applications, nécessite un déploiement dense des capteurs dans la ROI (région d'intérêt) qui est une phase décisive qui influe considérablement sur le fonctionnement et la performance du réseau. En effet, la couverture de la ROI est liée au déploiement, plus la zone est couverte mieux la surveillance est garantie, cependant de nombreux problèmes peuvent être engendrés après le déploiement, tel que l'apparition de capteurs redondants et des trous de couverture.

Ce présent mémoire vise à maximiser la couverture d'une région d'intérêt et à résoudre le problème de la relocalisation des capteurs redondants dans les réseaux de capteurs et de robots sans fil. À cet effet, nous avons proposé un nouvel algorithme de relocalisation pour déplacer les capteurs redondants à l'aide de robots mobiles, basé sur le principe de la loi de la réflexion de la lumière pour la marche des robots.

À propos de notre solution, les robots adoptent une marche aléatoire, la direction du déplacement sur l'axe dépend du calcul des probabilités en utilisant des formules mathématiques. L'adaptation de la marche aléatoire à notre solution est générée par le principe de la loi de la réflexion de la lumière qui permet au robot de découvrir les trous de couverture et les capteurs redondants autour de l'axe. Contrairement aux solutions proposées dans la littérature, qui sont de type centralisé ou qui utilisent une charge du robot égale à 1, notre solution est une solution distribuée qui permet de s'adapter à toute charge du robot (charge = N) et qui tiendra également compte de l'existence d'obstacles.

Dans la perspective de l'amélioration de notre travail, il serait possible d'utiliser un modèle en grille et de considérer un ensemble de robots déployés dans la zone d'intérêt.

Bibliographie

[1] : Somayya Madakam, R. Ramaswamy, Siddharth Tripathi, "Internet of Things (IoT): A Literature Review", India Article 2015.

[2] : altexsoft.com/blog/iot-architecture-layers-components ,Article consulté le 20/04/2021.

[3] : Ying Zhang , " Technology Framework of the Internet of Things and Its Application", Liaoning University of Technology, Conférence 2011.

[4] : M. C. V. Dr Ian F. Akyildiz, "Wireless Sensor Networks",Article 2010.

[5] : <http://www.lafabriquediy.com/tutoriel/liste-des-capteurs-229/> , consulté le 28/04/2021.

[6] : K. BEYDOUN, "Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs", Thèse de Doctorat en Informatique de l'université de FRANCHE-COMTE, 16 décembre 2009.

[7] : S.MERABET D.ZIANI,"Etude et simulation des techniques de minimisation de la consommation d'énergie dans les RCSFs", Thèse de MASTER en Télécommunication de l'université de Telemcen , 2013.

[8] : Ali JORIO, Le Clustering basé sur la classification spectrale pour l'optimisation d'énergie dans les réseaux de capteurs sans fil homogènes, Thèse de doctorat, UNIVERSITÉ MOHAMMED V, Rabat, 2015.

[9] : "Etude du RSSI pour l'estimation de la distaces dans les réseaux de capteurs sans fil", Thèse de Master en Informatique de l'université de Telemcen, 2017.

[10] : VANDERGRIFF, Linda J., et al. Nature and properties of light. Fundam. Photonics,Article 2008, p. 1-38.

[11] : Konrad Lorincz, Bor-rong Chen, Geoffrey Werner Challen, Atanu Roy Chowdhury, "A wearable sensor network platform for high-fidelity motion analysis",Conference NY, USA, 2009.

[12] : Jaime Lloret, Ignacio Bosch, Sandra Sendra, Arturo Serrano. " A wireless sensor network for vineyard monitoring that uses image processing",Article 2011.

[13] : E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, V.Z. Groza. "Sensor-based information appliances",Article Décembre 2000.

[14] : Mohamed-Ayoub Messous , "Supervision de l'opération d'irrigation des arbres fruitiers à l'aide des réseaux de capteurs sans fil", Thèse de MASTER en Informatique l'université de Blida.

- [15] : M.P. Durisic, Z. Tafa, G. Dimic, V. Milutinovic," A survey of military applications of wireless sensor networks". Computing ,Article juin 2012.
- [16] : Vehbi C Gungor and Gerhard P Hancke,"Industrial wireless sensor networks :Challenges, design principles, and technical approaches ", Article 2009.
- [17] : H.Yetgin, K.Cheung, M.El-Hajjar, and La Hanzo. A survey of network lifetime maximization techniques in wireless sensor networks,Article 2017.
- [18] : https://moodle.utc.fr/file.php/498/SupportWeb/co/Module_RCSF14,Article consulté le 24/04/2021.
- [19] : Bendjeddou Amira," Prolongation de la Durée de Vie des Batteries dans les Réseaux de Capteurs Sans Fil (RCSF)" ,thèse Doctorat en Informatique de l'université de Annaba 2014.
- [20] : S. Meguerdichian, F. Koushanfar, M. Potkonjak and M. B. Srivastava. "Coverage Problems in Wireless Ad-Hoc Sensor Networks". In 20th Annual Joint Conference of the IEEE Computer and Communications Societies, 3: 1380-1387, 2001.
- [21] : Y. Shang, W. Rumi, Y. Zhang and M. Fromherz," Localization from Connectivity in Sensor Networks" . IEEE Transactions on Parallel and Distributed Systems, 15(11):961-974,2004.
- [22] : wikipedia.org/wiki/Robot , Article consulté le 01 mai 2021.
- [23] : B.Desjardins, R.Falcon, R.Abielmona and E.Petriu. Reliable Robot-Assisted Sensor relocalisation via Multi-Objective Optimization.Artile 2016.
- [24] : LI, Qiao, NARASIMHAN, Venkat, et NAYAK, Amiya. Election-based sensor deployment and coverage maintenance by a team of robots. In: International Conference on Ad-Hoc Networks and Wireless. Springer, Cham, 2015. p. 163-177.
- [25] : Chien-Fu Cheng ,You-Cyun Chen and Jerry Chun-Wei Lin , " A CARRIER-BASED SENSOR DEPLOYMENT ALGORITHM FOR PERCEPTION LAYER IN THE IOT ARCHITECTURE,Artile 2020.
- [26] : JH.Ryu, M.Irfan, and A.Reyaz. A review on sensor network issues and robotics. Journal of Sensors, 2015.
- [27] : MOHAMED, Shaimaa M., HAMZA, Haitham S., et SAROIT, Iman Aly ,"Coverage in mobile wireless sensor networks (M-WSN)", 2017, vol. 110, p. 133-150.

- [28] : CHENG, Chien-Fu, CHEN, You-Cyun, et PAN, Meng-Shiuan. "The robot-assisted sensor deployment problem in Wireless Sensor Networks". In: 2017 International Conference on Applied System Innovation (ICASI). IEEE, 2017. p. 1868-1870.
- [29] : Mnasri, Sami and Nasri, Nejeh and Val, "Thierry The Deployment in the Wireless Sensor Networks: Methodologies, Recent Works and Applications. (2015)" In: International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks.
- [30] : F. Zhang, W. Guo, J. Yang, F. Yuan, L. Zhai, "Research on redundant channel model based on spatial correlation in IOT", in: Trustworthy Computing and Services, Springer, 2013, pp. 666–672.
- [31] : Sharma, V. Patel, R. Bhadauria, H. Prasad, "Deployment schemes in wireless sensor network to achieve blanket coverage in large-scale open area " A review. Egypt, Article 2016.
- [32] : DEBNATH, Sunandita, SINGH, Abhishek Kumar, et HOSSAIN, Ashraf. "A comprehensive survey of coverage problem and efficient sensor deployment strategies in wireless sensor networks" ,Indian Journal of Science and Technology,Article 2016.
- [33] : X. Liu , D. He ," Ant colony optimization with greedy migration mechanism for node deployment in wireless sensor networks", J. Netw. Comput. Appl. 39 (2014) 310–318.
- [34] : W.H.Liao , Y. Kao , YS. Li ," A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks ",Article 2011.
- [35] : M.R. Senouci , M. Abdelhamid , K. Assnour , "Localized movement-assisted sensor deployment algorithm for hole detection and healing",Article Mai 2014.
- [36] : LI, Xu, FLETCHER, Greg, NAYAK, Amiya, et al. Randomized "carrier-based sensor relocalisation in wireless sensor and robot networks. Ad hoc networks," vol. 11, no 7, p. 1951-1962 , 2013.
- [37] : LI, Xu, FLETCHER, Greg, NAYAK, Amiya, et al. Randomized "carrier-based sensor relocalisation in wireless sensor and robot networks". Ad hoc networks, 2013, vol. 11, no 7, p. 1951-1962.
- [38] : R. Soua , L.Saidane, P. Minet, "Sensors deployment enhancement by a mobile robot in wireless sensor networks" ,Article 2010.
- [39] : <https://www.camerecole.org/classes/83-la-reflexion-de-la-lumiere> , Article consulté le 27/06/2021.

[40] : <https://dataanalyticspost.com/Lexique/loi-des-grands-nombres> ,Article consulté le 26/06/2021.

[41] :Maxim A. Batalin et Gaurav S. Sukhatme, "The Analysis of an Efficient Algorithm for Robot Coverage and Exploration based on Sensor Network Deployment", Robotic Embedded Systems Laboratory Center for Robotics and Embedded Systems Computer Science Department, Université de Southern California, Los Angeles, USA