



HAL
open science

Optimisation multi-objectives d'une infrastructure réseau dédiée aux bâtiments intelligents

Mohamed Amin Benatia

► **To cite this version:**

Mohamed Amin Benatia. Optimisation multi-objectives d'une infrastructure réseau dédiée aux bâtiments intelligents. Intelligence artificielle [cs.AI]. INSA de Rouen, 2016. Français. NNT : 2016ISAM0024 . tel-01579381

HAL Id: tel-01579381

<https://theses.hal.science/tel-01579381>

Submitted on 31 Aug 2017

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

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



Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité INFORMATIQUE

Préparée au sein de « INSA-Rouen et CESI-Rouen »

Optimisation Multi-Objectives d'une Infrastructure Réseau Dédiée aux Bâtiments Intelligents

Présentée et soutenue par
Mohamed Amin BENATIA

Thèse soutenue devant le jury composé de	publiquement	le	13/12/2016
Mr. Farouk YALAOUI	PROFESSEUR DES UNIVERSITES / UT-Troyes	Rapporteur	
Mr. Nouredine ZERHOUNI	PROFESSEUR DES UNIVERSITES / ENSMM-Besançon	Rapporteur	
Mr. EL-HAMI Abdelkhalak	PROFESSEUR DES UNIVERSITES / INSA-Rouen	Directeur de thèse	
Mme. Anne LOUIS	ENSEIGNANT-CHERCHEUR (HDR) / CESI-Rouen	Co-directrice de thèse	
Mr. Nabil ABSI	PROFESSEUR DES UNIVERSITES / Ecole Nationale Supérieure des Mines de Saint-Etienne (ENSM-SE)	Examineur	
Mr. Abdallah EL-MOUNDI	PROFESSEUR DES UNIVERSITES / UT-Belfort-Montbéliard	Examineur	
Mr. David BAUDRY	ENSEIGNANT-CHERCHEUR (HDR) / CESI-Rouen	Encadrant	
Mr. M'hammed SAHNOUN	ENSEIGNANT-CHERCHEUR / CESI-Rouen	Encadrant	
Mr. Dimitri LEFEBVRE	PROFESSEUR DES UNIVERSITES / Université du Havre	Membre Invité	

Thèse dirigée par Abdelkhalak EL-HAMI et codirigée par Anne LOUIS, laboratoire LOFIMS et LINEACT-CESI



Remerciements

Cette thèse de doctorat complète ma scolarité, de ce fait nombreux sont les personnes que je souhaite remercier pour tout leur soutien et présence durant ce long chemin.

Je tiens à remercier mes directeurs de thèse Pr. Abdelkhalak EL-HAMI et Dr. Anne LOUIS pour leurs soutiens continus, leurs motivations et tous les conseils précieux qu'ils m'ont donnés. Je remercie également Dr. David BAUDRY qui m'a accueilli quand je suis arrivé au laboratoire et qui a toujours su m'orienter vers le bon chemin. Je tiens aussi à remercier le Dr. M'hammed SAHNOUN en qui j'ai trouvé plus qu'un encadrant, un ami. Avec lui j'ai appris et vécu beaucoup de choses pas seulement au niveau professionnel, mais aussi personnel.

Je tiens à remercier toute l'équipe du laboratoire LINEACT du CESI-Recherche pour tout. Les chercheurs permanents : Mourad MESSAADIA, Elodie PILLON, Nicolas BRIANT, Vincent HAVARD et Fabrice DUVAL ainsi que tous les doctorants : José ELOUNDOU et Sara MAHDIKHAH, qui ont toujours su répondre à mes Questions. Toutes nos discussions m'ont aidé à rédiger de meilleurs articles, à trouver de nouvelles pistes de recherche et même à faire des démonstrations avec lesquelles j'ai eu des difficultés pendant des semaines.

Je remercie également le directeur du département recherche du CESI Dr. Bélahcène MAZARI pour sa confiance en moi et les débats scientifiques auxquelles on était toujours conviés. Je voudrais aussi remercier Mme Christine TSAFACK, Assistante du Département Recherche du CESI de Rouen pour l'aide qu'elle nous a apporté dans l'édition et l'impression de ce document.

Pour finir, je remercie mes parents et toute ma famille, de m'avoir soutenu et encouragé pendant toutes mes études et surtout pendant ces années loin de la maison.

*À mes parents, mes frères et mon adorable femme,
À mes deux princesses Inès et Amina,
Je vous dédie cette modeste contribution.*

Résumé

Optimisation Multi-Objectifs d'une Infrastructure Réseau Dédiée au Bâtiment Intelligent

par Mohamed Amin BENATIA

Au cours de cette thèse, nous avons étudié le problème de déploiement des Réseaux de Capteurs Sans-Fil (RCSF) pour des applications *indoor* tel que le bâtiment intelligent. Le but de notre travail était de développer un outil de déploiement capable d'assister les concepteurs de RCSF lors de la phase de déploiement de ces derniers. Nous avons commencé cette thèse par la modélisation de tous les paramètres qui interviennent lors du déploiement des RCSF, à savoir : coût, connectivité, couverture et durée de vie. Par la suite, nous avons implémenté cinq algorithmes d'optimisation, dont trois multi-objectifs afin de résoudre le problème de déploiement. Deux cas d'études réelles (grande et petite instance) ont été identifiés afin de tester ces algorithmes. Les résultats obtenus ont montré que ces algorithmes sont efficaces quand il s'agit d'un petit bâtiment. Par contre, dès que la surface du bâtiment augmente les performances des algorithmes étudiés se dégradent. Pour répondre à cela, nous avons développé et implémenté un algorithme d'optimisation multi-objectifs hybride. Cet algorithme se base sur des notions de clustering et d'analyse de données afin de limiter le nombre d'évaluations directes qu'entreprennent ces méthodes pendant chaque itération. Afin d'assurer cette limitation d'évaluation les fonction de fitness sont approximées grâce aux réseaux de neurones et l'algorithme de classification *K-means*. Les résultats obtenus ont montré une très bonne performance sur les deux instances de tailles différentes. Ces résultats ont été comparés à ceux obtenus avec les méthodes classiques utilisées et sont compétitives et prometteuses.

Abstract

Multi-Objective Optimization of a Network Infrastructure Dedicated to Smart Building Applications

by Mohamed Amin BENATIA

In this thesis, we studied the Wireless Sensor Network deployment for indoor environments with a focus on smart building application. The goal of our work was to develop a WSN deployment tool which is able to assist network designers in the deployment phase. We begin this thesis with network modeling of all the deployment parameters and requirement, such as : cost, coverage, connectivity and network lifetime. Thereafter, we implement five optimisation methods, including three multi-objective optimization algorithms, to resolve WSN deployment problem. Then, two realistic study cases were identified to test the performances of the aforementioned algorithms. The obtained results shows that these algorithms are very efficient for deploying a small scale network in small buildings. However, when the building surface becomes more important the algorithms tends to converge to local optimum while consuming high processing time. To resolve this problem, we develop and implement a new Hybrid multi-objectif optimisation algorithm wich limits the number of direct evaluation. This algorithm is based on data-mining methods (Artificial Neural Networks and K-means) and tries to approximate the fitness value of each individual in each generation. At every generation of the algorithm, the population is divided to K clusters and we evaluate only the closest individual to cluster centroiide. The fitness value of the rest of population is approximated using a trained ANN. A comparative study was made and the obtained results show that our method outperformes others in the two sudy cases (small and big buildings).

Table des matières

Remerciements	i
Dédicaces	iii
Résumé	v
Abstract	vii
Table des matières	ix
Liste des Figures	xiii
Liste des Tableaux	xvi
Introduction	xviii
Contexte et motivation	xviii
Problématique	xviii
Objectifs	xix
Contributions de la thèse	xx
Organisation du manuscrit	xx
1 Bâtiment Intelligent et infrastructure réseau associée	1
1.1 Introduction	1
1.1.1 Intelligence ambiante et bâtiment intelligent	1
1.2 Laboratoires vivants (<i>Living-Labs</i>)	5
TRON House	5
Toyota Dream House (PAPI)	6
Le bâtiment Adream	7
L'appartement GERHOME	8
L'appartement DOMUS	9
L'appartement IRoom	9
1.2.1 Synthèse sur le bâtiment intelligent	11
1.3 Infrastructure réseau du bâtiment intelligent	11

1.3.1	Nœud capteur	12
1.3.2	Technologie de communication des RCSF	13
	Bluetooth-IEEE 802.15.1[1]	13
	Wibree[2]	13
	UWB-IEEE 802.15.3 [3]	13
	Wi-Fi	14
	Zigbee-IEEE 802.15.4[4]	14
1.3.3	Domaines d'applications des Réseaux de Capteurs Sans-Fil	15
	Le Militaire	16
	L'environnement	16
	Prévention contre les catastrophes naturelles	17
	La santé	17
	Le bâtiment intelligent	18
1.3.4	Limites des réseaux de capteurs sans-fil	18
1.4	Outil d'aide au déploiement d'un RCSF	19
1.4.1	Architecture de l'outil de déploiement des RCSF	19
	Collecte des exigences	20
	Conception du réseau	21
	Affichage des résultats	21
1.5	Conclusion	21
2	Modélisation de la problématique et méthodes de résolution	23
2.1	Introduction	23
2.2	Modélisation des critères de déploiement	24
2.2.1	Espace de déploiement	24
2.2.2	Connectivité	24
	2.2.2.1 Étude de l'existant	26
	Modèle de FRIIS (espace-libre)	28
	Modèle Line Of Sight (LOS)	28
	Modèle Multi-Wall (MWM)	30
	Modèle à lancer de rayons	31
2.2.3	Détection des évènements par le RCSF	31
	2.2.3.1 Modèle de Détection Binaire (MDB)	33
	2.2.3.2 Modèle de Détection Probabiliste (MDP)	34
2.2.4	Coût de déploiement	36
2.2.5	Durée de vie du réseau	36
	2.2.5.1 Communication directe	38
	Sans sur-écoute	38
	Avec sur-écoute	39
	2.2.5.2 Communication multi-sauts	39
	Modèle linéaire :	39
	Modèle non-linéaire	39
2.3	Modèles retenus et expérimentations	44
	2.3.1 Préliminaires et hypothèses	44
	2.3.2 Codage de la solution	45
	2.3.3 Coût du RCSF	46
	2.3.4 Couverture	46

2.3.5	Connectivité	48
2.3.6	Durée de vie	48
2.4	Expérimentations	49
2.4.0.1	Capteurs retenus	49
2.4.1	Puissance de transmission vs. distance	51
2.4.1.1	Cas d'étude	51
2.4.1.2	Modèle de propagation de FRIIS	51
	Résultats	52
2.4.1.3	Modèle Multi-Wall vs. Modèle de FRIIS	56
2.5	Conclusion	59
3	Méthodologie de déploiement du réseau de capteurs sans-fil	61
3.1	Introduction	61
3.2	Résolution du problème de déploiement des RCSF	62
3.2.1	Méthodes déterministes	62
3.2.1.1	Méthode de gradient	63
3.2.1.2	Méthode du simplexe	64
3.2.1.3	Branch & Bound	64
3.2.1.4	Branch & Cut	64
3.2.2	Méta-heuristiques	65
3.2.2.1	Recherche Tabou	67
3.2.2.2	Recuit Simulé	67
3.2.2.3	Algorithmes Évolutionnaires	68
3.2.2.4	Intelligence en Essaim	69
3.2.3	L'optimisation dans le problème de déploiement des RCSFs	70
3.3	Architecture proposée	74
3.3.1	Méthodes utilisées	76
3.3.1.1	Algorithme génétique (AG)	76
3.3.1.2	SPEA-II	77
3.3.1.3	PESA-II	78
3.3.1.4	NSGA-II	78
3.3.1.5	Differential Evolution	79
3.3.2	Adaptation des algorithmes au problème de déploiement du RCSF	80
3.3.2.1	Codage du chromosome	81
3.3.2.2	Création de la population initiale	81
3.3.2.3	Sélection des individus	82
3.3.2.4	Opérateur de croisement	82
3.3.2.5	Opérateur de mutation	83
3.3.2.6	Opérateur de réparation	83
3.3.3	Métriques de comparaison	84
3.4	Résultats et discussion	86
3.4.1	Modélisation Mathématique	86
3.4.2	Expérimentation I (Petite instance)	88
3.4.2.1	Cas d'étude	88
3.4.2.2	Paramétrages des opérateurs	89
3.4.2.3	Algorithme génétique vs. NSGA-II	91
3.4.2.4	Validation et discussion	98

3.4.2.5	Comparaison avec d'autres Variantes d'algorithme évolutionnaires	98
3.4.3	Expérimentation II	101
3.5	Conclusion	108
4	Méta-heuristique hybride pour le déploiement des RCSF	109
4.1	Introduction	109
4.2	Analyse des temps consommés	110
4.3	Hybridation dans les méta-heuristiques	112
4.3.1	Problèmes de conception	112
	Classification hiérarchique	113
	Classification horizontale	114
4.3.2	Problèmes d'implémentation	115
4.4	Architecture hybride proposée	116
4.4.1	Approximation de la valeur de fitness	116
4.4.2	Analyse de l'existant	117
4.4.3	Algorithme proposé	119
4.4.4	Limiter les évaluations	121
4.4.4.1	Algorithme <i>K-means</i>	122
4.4.4.2	Réseau de Neurones	123
4.4.4.3	Combinaison de la fonction de fitness	125
4.5	Résultats et discussion	125
4.5.1	Expérimentation-I (Petite Instance)	126
4.5.2	Expérimentation-II (Grande Instance)	128
4.6	Outil de déploiement des RCSF	131
4.7	Conclusion	134
5	Conclusion et Perspectives	135
	Bibliographie	142

Table des figures

1.1	De l'ordinateur à l'intelligence ambiante	2
1.2	Utilisation des TIC dans le monde [5]	3
1.3	Utilisation d'Internet dans le monde [5]	3
1.4	Différentes prises de vue de la TRON House	6
1.5	Toyota Dream House (PAPI)	7
1.6	Façade et toit photovoltaïque du bâtiment Adream	8
1.7	Architecture et infrastructure réseau de l'appartement GERHOME	9
1.8	Plan général du laboratoire	10
1.9	Plateforme Domus	10
1.10	Vue générale de l'appartement IRoom	10
1.11	Architecture d'un réseau de capteurs sans-fil	12
1.12	Anatomie d'un noeud capteur	12
1.13	Panorama des applications d'un RCSF	16
1.14	Principe de fonctionnement de la collecte des verres	17
1.15	Système d'aide aux non-voyants	18
1.16	Différents outils d'aide au déploiement de réseau LAN	20
1.17	Différentes phases de l'outil d'aide au placement	20
2.1	Plan du bâtiment discrétisé en cellule	25
2.2	Deux antennes séparées par une distance R [6]	29
2.3	Diagramme de rayonnement 3-D en sphère d'une antenne isotrope[6]	29
2.4	Lien radio typique LOS[7]	30
2.5	Quelques modèles de propagation à lancer de rayons	32
2.6	Modèle de Détection Binaire [8]	33
2.7	Modèle de Détection Probabiliste [9]	35
2.8	Modèle du coût de déploiement	37
2.9	Processus de reporting de données dans des RCSF opérant sous Zigbee	41
2.10	Variables utilisées [10]	42
2.11	Processus de retranscription Grille-Solution	45
2.12	Modèle de couverture d'évènement	47
2.13	Capteurs implémentés dans le Living-Lab du CESI	50
2.14	Architecture générale du bâtiment	51
2.15	Puissance de transmission vs. Distance	52
2.16	Énergie consommée par chaque noeud du réseau	53
2.17	Topologie réseau du deuxième scénario	54
2.18	Énergie consommée par chaque noeud du réseau (premier et deuxième scénario)	55
2.19	Kit de développement Texas Instruments (TI)	57

2.20	Puissances de réception dans le bâtiment	58
2.21	Énergie consommée par chaque nœud	58
3.1	Classification des différentes méthodes d'optimisation [11]	63
3.2	Classification des méthodes déterministes[12]	63
3.3	Division du problème F en sous-problèmes	65
3.4	Exemple de méthodes stochastiques[12]	66
3.5	Architecture d'un algorithme évolutionnaire [13]	70
3.6	Architecture du système de déploiement du RCSF	75
3.7	Architecture de l'algorithme NSGA-II	79
3.8	Exemple d'une solution (individu)	81
3.9	Exemple de croisement entre deux parents en un point	83
3.10	Exemple de mutation	84
3.11	Interface graphique du questionnaire	85
3.12	Vue 3D de l'espace de déploiement	89
3.13	Variation de la probabilité de mutation	90
3.14	Variation de la probabilité de croisement	91
3.15	Impact de la variation de la taille de la population sur la valeur de fitness	92
3.16	Évolution de la valeur de fitness)	93
3.17	Solution issue de la première génération (gauche), et solution de la dernière génération (droite)	93
3.18	Taille de la population initiale vs. temps d'exécution	95
3.19	Front de Paréto de l'algorithme NSGA-II	96
3.20	Capteurs et routeurs déployés selon la solution proposée par NSGA-II	97
3.21	Front de Pareto obtenu pour le placement de noeuds routeurs	97
3.22	Front de Pareto obtenu pour le placement de noeuds capteurs	97
3.23	résultat de la simulation sur NS-2 (données traitées par Visual Trace Analyser)	99
3.24	Évolution de la valeur de fitness vs. Génération des algorithmes	100
3.25	Évolution de la valeur de fitness vs. temps d'exécution	100
3.26	temps d'exécution cumulé dans chaque génération	101
3.27	Valeur de Fitness VS. Génération (déploiement capteurs)	102
3.28	Temps consommé dans les génération de l'algorithme (déploiement capteurs)	102
3.29	Plan du bâtiment avec les emplacements des prises	103
3.30	Valeurs d'atténuations radio des matériaux selon[14]	104
3.31	Valeurs d'atténuations radio des matériaux dans le laboratoire IRISE du CESI	104
3.32	Différentes vues de l'interface de l'outil SmartRF-Studio (TI)	105
3.33	Évolution de la valeur de fitness vs. Génération des algorithmes (Grande instance)	106
3.34	Évolution de la valeur de fitness (somme pondérée) dans le temps d'exécution (Grande instance)	107
3.35	temps d'exécution cumulé dans chaque génération (Grande instance)	107
4.1	Temps consommé par les phases : croisement, classement et mutation	111
4.2	Temps consommé par la phase d'évaluation de l'algorithme	112
4.3	Classification des métaheuristiques hybrides [15]	113
4.4	Architecture générale du bâtiment	114

4.5	Exemples d'hybridation Haut-niveau	115
4.6	Architecture de l'algorithme proposé	121
4.7	Schéma d'un neurone biologique (adapté de [16])	123
4.8	Modèle d'un neurone artificiel [17]	124
4.9	Erreur quadratique entre les valeurs de fitness estimée et réelle	127
4.10	valeur de fitness vs génération de l'algorithme(déploiement capteurs)	127
4.11	valeur de fitness vs génération de l'algorithme (déploiement routeurs)	128
4.12	Fonctions objectives normalisées	128
4.13	valeur de fitness vs génération de l'algorithme	129
4.14	Temps cumulé dans chaque génération de l'algorithme (déploiement cap- teurs)	129
4.15	Valeur de fitness vs génération de l'algorithme	131
4.16	Temps cumulé par génération de l'algorithme	131
4.17	Temps cumulé par génération de l'algorithme (déploiement capteurs)	132
4.18	Interface principale de l'outil de déploiement des RCSF	132
4.19	Vue 3D de la propagation radio	133
4.20	Vue 3D de la couverture des capteurs déployés	133

Liste des tableaux

1.1	Tableau comparatif entre les différents standards de communications [18]	15
2.1	Tableau comparatif des modèles de propagation	32
2.2	Tableau récapitulatif des capteurs utilisés	50
2.3	Puissance de transmission vs. Distance	52
2.4	Récapitulatif des paramètres utilisés dans NS 2.35	53
2.5	Énergies consommées par chaque noeud résultat NS 2.35 (scénario-1)	54
2.6	Énergies consommées par chaque noeuds (deuxième scénario)	55
3.1	rasumé des travaux qui ont utilisé de l'optimisation Multi-Objectif des RCSFs	71
3.2	Exemple d'une population initiale de cinq (05) individus	82
3.3	Symboles utilisés	87
3.4	Valeurs des paramètres utilisés	89
3.5	Impact de la variation des probabilités de mutation et de croisement sur le temps d'exécution (secondes)	90
3.6	Impact de la variation des probabilités de mutation et de croisement sur la valeur de fitness	90
3.7	Valeurs des différents objectifs de la solution optimisée retenue	92
3.8	Impact de la taille de population sur le temps d'exécution	94
3.9	Positions des nœuds routeurs (Sink)	94
3.10	Positions des nœuds capteurs	95
4.1	Travaux sur l'approximation de la valeur de fitness par des réseau de neurones artificiels	120

Introduction

Contexte et motivation Avec la demande croissante en terme de consommation d'énergie dans le secteur résidentiel et grâce aux nouveaux systèmes émergents tels que les réseaux de capteurs sans fil; de nouvelles solutions visant à garantir une faible consommation d'énergie et un confort pour l'habitant/l'utilisateur ont vu le jour, c'est le cas des bâtiments intelligents.

Selon [19] le bâtiment intelligent est défini comme un immeuble doté de la capacité d'interaction avec son environnement. Ce dernier effectue une collecte d'informations liées à son environnement physique (température, luminosité, présence, ...), afin d'entreprendre par la suite des décisions (entretenir des actions) tels que : l'extinction des lumières, l'ouverture/fermeture des fenêtres (ou portes) dans le but d'avoir une meilleure efficacité énergétique du bâtiment. La solution naturelle à cela, serait la mise en place d'une technologie dotée de ces capacités de collecte et de communication des données. La technologie des Réseau de Capteurs Sans Fil (RCSF) autonome en énergie a connu un grand succès dans ce domaine.

En général, un RCSF est constitué d'un ensemble de composantes appelées nœuds capteurs et dotées d'une capacité de collecte d'informations sur des grandeurs physiques qui peuvent changer selon les applications (température, humidité, présence, ...). Ces nœuds recueillent les informations et envoient des rapports périodiques à un nœud spécifique appelé nœud puits (routeur ou Sink en Anglais). Ce nœud est considéré comme étant une passerelle entre le champ de captage (composé par les différents nœuds capteurs éparpillés dans une région) et le système de gestion du bâtiment (serveur, internet ...).

Problématique Les principales problématiques de ce type de réseau à savoir : la consommation énergétique, couverture du réseau, coût de déploiement, ajoutée à celle de l'application qui nécessite une couverture maximale du bâtiment et une durée de vie (du réseau) de plusieurs années, impose une excellente conception du réseau. De nombreuses études [20–23] montrent qu'un bon déploiement du RCSF peut assurer, non seulement, une durée de vie maximale, mais aussi, une bonne couverture réseau [24] .

Le déploiement d'un RCSF consiste à élire les positions des différents nœuds capteurs/-routeurs qui composent le réseau [25]. Plusieurs approches ont favorisé par le passé un déploiement manuel et une approche expérimentale où le concepteur déploie le réseau et calcule certaines métriques par la suite. Si le réseau déployé est fiable les positions des différents nœuds sont validées, sinon le concepteur change les positions des nœuds et reprend la même démarche jusqu'à ce qu'il trouve un réseau qui satisfait ces attentes. Cette approche est coûteuse et demande énormément de temps et de ressources (personnes).

Avec l'avancée technologique et le développement des ordinateurs qui sont devenus très performants et dotés d'une plus grande capacité de mémoire et de calcul, des outils de déploiement sont apparus. Grâce à ces derniers, les concepteurs seront assistés par des machines (systèmes) qui proposent la topologie optimale du réseau en utilisant des notions de propagation radio et de mathématiques.

Des outils de déploiement et de conception de RCSF, utilisant des notions d'optimisation et de méta-heuristiques, ont vu le jour dans la dernière décennie [26–30]. À part quelques travaux, ces solutions proposées traitent une ou deux contraintes au maximum. Le développement d'un outil capable de traiter les trois objectifs prédéfinis est une nécessité primordiale dans le domaine de conception d'un RCSF. Pour cela, une bonne modélisation du système et un bon choix de la méthode d'optimisation doivent être examinés.

Objectifs L'objectif principal de la thèse est de proposer un outil capable d'aider les concepteurs de RCSF lors de la phase de déploiement. Pour pouvoir développer un tel outil, une bonne modélisation du système est nécessaire. Aussi, une phase d'optimisation multi-objectifs doit être considérée. L'approche proposée consiste à combiner des notions de communication sans-fil, de réseau, de modèles fonctionnels et d'optimisation multi-objectifs afin d'avoir une topologie réseau.

Cette approche vise à assurer un déploiement optimal du RCSF sous les contraintes suivantes :

- La connectivité : le réseau doit assurer une bonne qualité de service. Pour cela, les différents nœuds du réseau doivent pouvoir transmettre/recevoir parfaitement les données, sans erreur et avec le minimum de perte possible.
- La couverture : le réseau doit couvrir la totalité du bâtiment pour qu'on puisse avoir une vue globale et exacte des phénomènes physiques qui s'y produisent.
- La durée de vie du réseau : le réseau doit avoir une durée de vie de plusieurs années, ce qui réduit les coûts de maintenance et assure une fiabilité du système implémenté.
- Le coût : le réseau installé doit avoir un coût de déploiement minimal pour pouvoir toucher le plus grand public.

Contributions de la thèse Les contributions globales de ce travail sont les suivantes :

La première contribution consiste à proposer une modélisation du RCSF ainsi que l'environnement de déploiement (architecture du bâtiment). Cette contribution comprend la modélisation des communications entre les différents nœuds du réseau ainsi que les phénomènes de propagations d'ondes radio. Ainsi, une modélisation de la consommation énergétique du réseau sera proposée.

La deuxième contribution consiste à exprimer toutes les contraintes de placement, ainsi que celles liées à la transmission grâce à un modèle pouvant être exploité lors de la phase d'optimisation multi-objectif.

La troisième contribution majeure consiste à résoudre le problème de déploiement optimisé du RCSF par une méthode d'optimisation multi-objectifs. Plusieurs méthodes s'offrent à nous tels que : MOGA (Multi-Objective Genetic Algorithm), NSGA-II (Non-Dominated Sorting Genetic Algorithm), MPSO (Multi-Objective Particle Swarm Optimization) . . . Afin de les valider, une adaptation de notre problème à ces méthodes, ainsi qu'une étude comparative entre ces dernières ont été proposées.

En menant les expérimentations, nous avons remarqué une lenteur d'exécution lors de l'utilisation de ces méthodes sur des bâtiment de grande taille. Afin de répondre au problème pouvant être posé par les bâtiments de grandes tailles nous proposons une méthode d'optimisation hybride basée sur des concepts d'analyse de données et de data-mining. Nous avons utilisé l'algorithme *K-means* afin de diviser la population en plusieurs sous-parties. Une fois la population divisée, nous n'évaluerons que les individus les plus proches des centroides de chaque classe. Les valeurs de fitness du reste de la population sont approximées grâce aux réseaux de neurones.

Enfin, la dernière contribution sera le développement d'un système (outil), facile à utiliser, et qui prend en entrée l'architecture du bâtiment pour pouvoir, par la suite, donner les emplacements optimaux des nœuds capteurs. Pour cela, une batterie de test de fiabilité du système (tolérance aux pannes) et de vérification est intégrée.

Organisation du manuscrit Le manuscrit est organisé en quatre chapitres. Dans le premier, un état de l'art sur les bâtiments intelligents, les infrastructures réseaux associées et les outils de déploiement existants; ainsi qu'une brève introduction au problème de déploiement de ces infrastructures réseau est présenté. Dans le deuxième chapitre, nous présenterons les différents modèles de RCSF existants. Nous exposerons par la suite les modèles retenus ainsi que les résultats des nos premières expériences. Dans le Chapitre 3, nous aborderons la notion d'optimisation mono et multi-objectives. Nous présenterons par la suite la méthodologie de déploiement adoptée. Les résultats d'expérimentations menés sur des cas d'études réelles seront exposés. Dans

le chapitre quatre, nous présenterons l'architecture de l'outil de déploiement proposée. Notre travail se termine par une conclusion où l'essentiel de notre étude est présenté ainsi que les perspectives que nous jugeons bénéfiques à notre travail.

Chapitre 1

Bâtiment Intelligent et infrastructure réseau associée

1.1 Introduction

1.1.1 Intelligence ambiante et bâtiment intelligent

D'après [31] la notion de confort dans l'habitat date des années 80 avec l'introduction des appareils électroménagers et l'automatisation de certaines tâches ménagères. En effet, la domotique vient du mot latin "Domus" qui veut dire Maison et "-tique" désignant l'automatique. Cette notion a été introduite vers la fin du 20^{ème} siècle dans l'objectif de faciliter les tâches ménagères en automatisant certains processus et de rendre les différents objets de la maison communicants tels que : machine à laver, réfrigérateur, lave vaisselle . . .

Avec l'introduction des appareils mobiles communicants vers la fin du 20^{ème} siècle un nouveau concept a vu le jour : l'intelligence ambiante (Informatique Ubiquitaire) comme on peut le voir sur la Figure 1.1. Selon [32] l'intelligence ambiante est définie comme : *Un environnement numérique qui, de manière proactive, mais sensible, soutient les gens dans leur vie quotidienne.* Elle intègre toutes les technologies de l'apprentissage automatique, des logiciels basés sur les agents et de la robotique. De nos jours, les appareils nomades communicants font partie de notre vie quotidienne. Selon les données de l'UIT (l'institution spécialisée des Nations Unies pour les technologies de l'information et de la communication (TIC)), la majorité des personnes disposent d'un téléphone mobile intelligent (Smart Phone) doté d'un processeur très puissant, qui peut être connecté à internet, et capable de communiquer (s'interfacer) avec d'autres appareils (Télévision, tablette tactile, voiture . . .). Comme on peut le voir sur la Figure 1.2, l'utilisation des TIC

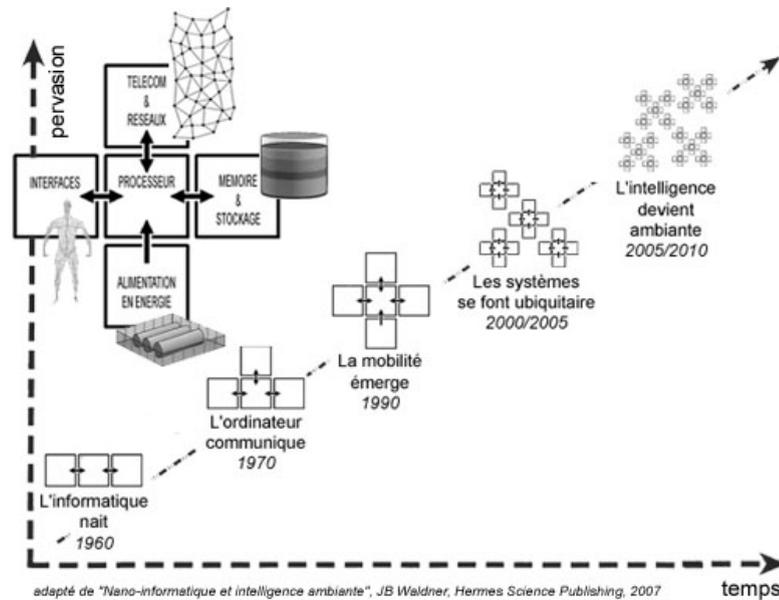


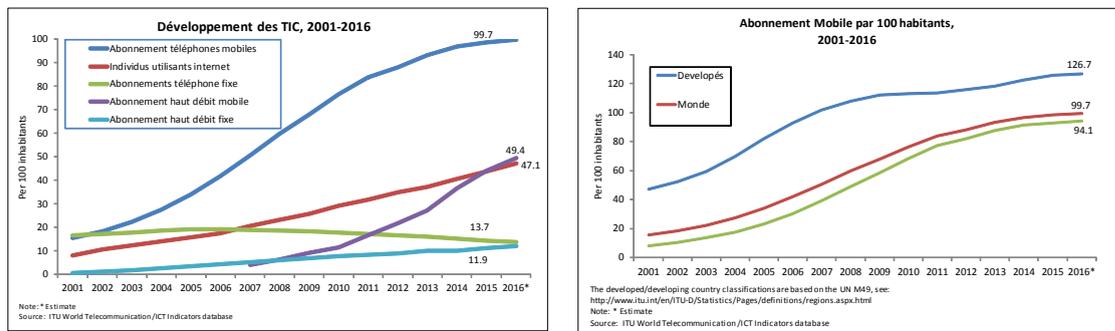
FIGURE 1.1: De l'ordinateur à l'intelligence ambiante

(Technologies de l'Information et de la Communication) ne cesse d'augmenter d'année en année. On compte, en 2016, 99.7 % de personnes ayant souscrit un abonnement téléphonique (1.2b) et 47.1% utilisant internet dans le monde (79.1% en europe 1.3b). C'est grâce à cette avancée technologique qu'ont émergé plusieurs nouveaux concepts tels que : les VANETs (Vehicular Ad hoc NETworks)[33], Bâtiment Intelligent [34], Smart Grids [35]...

Nous nous intéressons dans cette thèse aux différents aspect du bâtiment intelligent (BI), ou Smart Building. Un bâtiment intelligent intègre une multitude d'objets dotés de capacités de communication et d'échange d'informations liées à leurs environnements physiques (lumière, humidité, température, consommation d'énergie ...). Aussi, le BI peut prendre des décisions visant à apporter plus de confort, grâce à l'interaction entre bâtiment et usager et le rapprochement de l'humain de sa résidence. Ces décisions peuvent être centrées sur l'humain (confort) ou bien sur l'efficacité énergétique du bâtiment.

D'après Gallissot les débuts de la domotique remontent au début du 20^{ème} siècle. Dès les débuts de la domotique, un défi a été fixé qui représente l'intégration des nouvelles technologies au sein du domaine traditionnel qu'est le bâtiment. Gallissot liste plusieurs maisons qui se voulaient innovatrices, construites dans les années 80, on peut en citer quelques unes : "Lyon Panorama" à Caluire (Rhône), et DELTA à Bonneville (Haute-Savoie). Aussi, des structures expérimentales se sont développées parallèlement telles que : HD2000 à Rennes et SED à Saint-Rémy-Lès-Chevreuse.

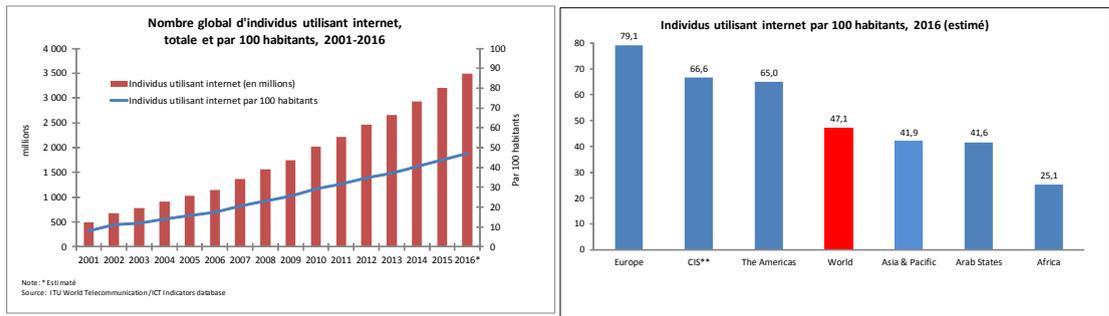
Au même temps, plusieurs démonstrateurs étaient conçus dans le monde tel que la "TRON Intelligent House (1989)" conçue par Sakamura. Cette maison servait de démonstrateur



(A) Développement des TIC

(B) Abonnement téléphonie mobile dans le monde

FIGURE 1.2: Utilisation des TIC dans le monde [5]



(A) nombre d'individus ayant utilisés internet

(B) utilisation d'internet dans le monde

FIGURE 1.3: Utilisation d'Internet dans le monde [5]

pour la technologie TRON, appliquée dans le domaine du bâtiment.

Après cette première expérience, plusieurs autres constructions ont été réalisées, telles que la Toyota Dream House PAPI (2010). Les applications développées dans ces démonstrateurs étaient très orientées usage (ex. : système de rangement intelligent).

Récemment, les plateformes sont développées autour d'une thématique précise. Nous pouvons citer les travaux de Thomesse, ou de Nourizadeh et al. menés sur le maintien à domicile des personnes dépendantes. Une autres thématique intéressante est l'efficacité énergétique des bâtiments. Nous pouvons citer les travaux de Clanché et al. qui portent sur le développement de système de gestion de l'énergie.

De nos jours, les plates-formes développées allient humain et bâtiment avec le concept de *Living-Lab*. Ces derniers mettent en oeuvre des écosystèmes propices aux échanges pluridisciplinaires (ex. : l'appartement domus). Selon [36] un bâtiment intelligent doit relever plusieurs défis qui se présentent comme suit :

- L'information ne peut pas circuler librement de l'intérieur de la maison vers le monde extérieur et vice-versa.

- La maison fonctionne avec des outils informatiques intégrés qui ne sont pas capable de communiquer entre eux.
- Son automatisation consiste en un assemblage de plusieurs ”gadgets” (dispositifs) tels que les appareils électroménagers.
- Les fonctions intégrées dans cet habitat sont sophistiquées et difficiles à utiliser.

Ces défis peuvent être résumé en :

- Le premier défis consiste à intégrer au sein de l’habitat une infrastructure (technique) capable de communiquer avec le monde extérieur.
- Le second défis est le critère d’ergonomie et de simplicité d’utilisation. Le bâtiment doit avoir une interface, liant bâtiment et usager, qui doit être intuitive et facile à comprendre et à utiliser.

D’autre part, Brun et Descamp [37], ont insisté sur la minimisation des coûts de construction du bâtiment intelligent. Ils proposent la mise en place de formations pour les artisans devant construire le bâtiment. Dans la vision de Brun et Descamp du bâtiment intelligent, l’architecte qui prend en charge la construction du bâtiment doit être bien formé et prend une grande part de responsabilité dans le bon fonctionnement du bâtiment.

Dans leurs publication Edwards et al. [38], listent sept défis qu’un habitat intelligent doit relever. Ces défis sont comme suit :

- Faciliter la compréhension des appareils intelligents présents dans l’habitat, en offrant un aperçu de ce que ces dispositifs peuvent faire, ce qu’ils ont fait et comment nous allons les contrôler ;
- Veiller à ce que l’avenir de la maison intelligente ne soit pas un ensemble d’ilots isolés de fonctionnalités incompatibles, mais plutôt celui dans lequel l’occupant peut s’attendre à ce que les systèmes présents dans sa maison travaillent ensemble et de manière fluide ;
- Veiller à ce qu’il n’existe pas d’administrateur du système ;
- Prêter attention aux routines stables et contraignantes de la maison, plutôt que les facteurs externes, y compris les capacités de la technologie elle-même ;
- Etre conscient des effets plus larges du travail de conception de la maison intelligente, et de réaliser que même les technologies les plus simples telles que la machine à laver peuvent avoir de grandes conséquences sur sur la dynamique de la maison et de la société elle-même ;
- Veiller à ce que le système soit fiable ;
- Créer des systèmes qui garantissent que les usagers comprennent aussi bien la pragmatique de capteurs, l’interprétation et l’action des machines et des dispositifs présents dans leurs bâtiments.

Ces défis peuvent être classés en deux catégories : besoins techniques et besoins humains.

- Besoins techniques : Chaque habitat intelligent doit résoudre tous les défis reliés à son aspect technique tel que : l’interopérabilité, fiabilité, sécurité, ainsi que la capacité de fonctionner en présence d’ambiguïté).
- Besoins humains : Un habitat intelligent doit être facile à appréhender par son habitant, en effet, chaque fonctionnalité qu’intègre cet habitat doit être facile à comprendre et à utiliser, mais aussi, elle doit avoir une interface attractive (ergonomie adaptée à un environnement domestique). Finalement, l’habitat intelligent doit respecter le contexte social des personnes.

Comme pour les critères vus en [38], Chan et al. [39] proposent des défis que l’habitat intelligent doit résoudre pour des applications liées à l’aide à la personne et à la télémédecine. Les défis posés par Chan et al. portent surtout sur des critères d’utilisation (ergonomie, besoins, acceptabilité...). Mais aussi, les critères technologiques qui sont : la fiabilité et la fonctionnalité des dispositifs embarqués sur les personnes (capteurs), et le bon fonctionnement des applications dédiées à la surveillance de la personne.

1.2 Laboratoires vivants (*Living-Labs*)

Dans ce contexte des *Living-Labs*, nous présentons quelques bâtiments intelligents réalisés en collaboration entre des laboratoires et des industriels qui servent, comme banc d’essai pour les travaux de recherches.

TRON House La construction de cette maison a débuté vers Octobre 1988 et s’est achevée vers l’été de l’année 1989 (Juillet 1989) avec un coût de un Milliard 290 millions d’euros. Plusieurs compagnies ont pris part au projet (15 au total), afin de construire une maison témoin propriété de Nippon Homes Corporation. Le projet a reçu les critiques des médias et des journaux qui disaient que cette maison coûtait trop chère, ou bien, cette maison était considérée comme hantée (les portes et les fenêtres s’ouvrent toutes seules, les feux se déclenchent automatiquement ...).

Le but de Sakamura était de réunir et fusionner ”Humains, nature et ordinateur”, pour cela, 380 ordinateurs (capteurs, activateurs, ordinateurs ...), tous interconnectés via le système TRON Architecture (mis au point par Ken Sakamura), ont été déployés dans la maison.

Cette maison est restée opérationnelle pendant trois années, elle représentait un banc d’essai pour les systèmes informatiques. Après les critiques de la presse, la maison a été totalement démontée en 1992. De nos jours, la maison a été transformée en un ensemble



(A) Face avant de la TRON House



(B) Face arrière de la TRON House

FIGURE 1.4: Différentes prises de vue de la TRON House

de bureaux.

Toyota Dream House (PAPI) Le projet *Toyota Dream House* a été proposé par la firme automobile *Toyota* en collaboration avec le Pr. Sakamura[40]. Toyota a demandé au Pr. Sakamura de lui construire une maison intelligente qui s'accorde avec les coutumes locales (architecture locale des maisons).

Le Pr. Sakamura a poussé son savoir-faire, en se basant sur l'ancien projet *TRON House* et a intégré dans la nouvelle maison plus d'objets communicants. La *Toyota Dream House* a été conçue pour être interfacée avec d'autres technologies *Toyota* telles que les voitures électriques. Dans cette thématique, *Toyota* voulait intégrer sa plus importante technologie qui est : Motor Corporation's hybrid sedan à la maison pour que les voitures électriques puissent être rechargées par le circuit électrique de la maison. D'un autre côté, en cas de catastrophe naturelle (tremblement de terre), ou en cas de coupure d'électricité, la maison puisera dans les batterie des véhicules électriques qui peuvent fournir 36 heures d'autonomie à la maison.

Cette maison appelée PAPI ou *Toyota Dream House* est totalement différente de l'ancienne (*TRON House*). Car, en 1988 la technologie des réseaux sans-fil n'existait pas encore, ce qui nécessitait la mise en place d'un large réseau filaire pour interconnecter les différents dispositifs (capteurs, ordinateurs, ...) du bâtiment.

Ce nouveau projet, d'une superficie de $689m^2$, s'appuie sur les concepts d'internet des objets (*IoT : Internet of Things*) et d'intelligence ambiante. L'habitant dispose ainsi d'une interface (contextuelle) de gestion de l'habitat, ce qui favorise l'interaction homme machine imaginée par le Pr. Sakamura. Par ailleurs, plusieurs autres technologies sans-fil ont été implémentées telles que : RFID, capteurs sans-fil. ... Cette maison est toujours en activité.

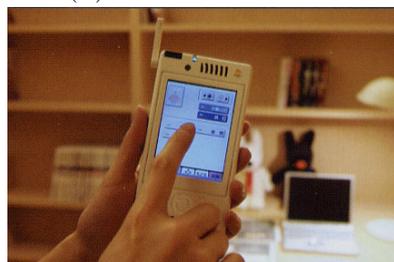
Pour permettre à un grand nombre de chercheurs de mener à bien leurs travaux dans le domaine du bâtiment intelligent, plusieurs laboratoires ont entrepris des travaux dans



(A) entrée du bâtiment



(B) Véhicules électriques



(C) Commande électronique

FIGURE 1.5: Toyota Dream House (PAPI)

le but de reproduire des espaces habitables intelligents. Nous pouvons classer ces plateformes selon leurs thématiques de recherche. Nous pouvons citer les travaux menés au *LAAS-CNRS* [41] qui portent sur l'efficacité énergétique du bâtiment grâce à l'utilisation des énergies renouvelables tels que les panneaux solaires. Une autre thématique émergente est le maintien à domicile des personnes dépendantes. Dans ces projets on vise à redonner une vie autonome, dans leurs domiciles, à des personnes souffrant de diverses pathologies et handicaps qui devraient normalement les contraindre à une hospitalisation ou à un placement en institution spécialisée. On peut citer dans cette thématique, les projets menés dans l'appartement *GERHOME* à Sophia Antipolis en collaboration entre le CHU de Nice [42], le CSTB, ainsi que l'INRIA [42]. Ces laboratoires d'expérimentation *in-vivo* (avec des personnes vivant à l'intérieur) sont des plateformes de développement des nouvelles technologies de la communication centrées sur l'utilisateur.

Le bâtiment Adream La construction du bâtiment Adream a débuté en Juin 2010 à l'université de Toulouse et a été livré en Décembre 2011. Le projet a été financé par le contrat Etat-Région 2007-2010 et regroupait quatre partenaires : Union européenne (45%), Région Midi-Pyrénées (35%), Communauté urbaine du Grand Toulouse (14%), et le CNRS (7%). L'objectif principal du projet est de mettre en œuvre une méthodologie ainsi que les technologies nécessaires aux traitements des problèmes d'hétérogénéité, d'interopérabilité des systèmes, de sécurité, et surtout d'autonomie et de gestion de l'énergie dans le bâtiment.

Le bâtiment est construit sur une surface de $1700m^2$ et dispose de plusieurs panneaux



FIGURE 1.6: Façade et toit photovoltaïque du bâtiment Adream

photovoltaïques d'une superficie de $720m^2$ couvrant toute la toiture et la façade. Le bâtiment est composé de bureaux qui occupent $700m^2$ de la surface totale du bâtiment. Plusieurs nouvelles technologies ont été intégrées au bâtiment telles que les capteurs de : sensoriels (présence), luminosité, consommation d'énergie, . . . La Figure 1.6 représente la façade extérieure du bâtiment *Adream*.

L'appartement GERHOME En collaboration entre le CSTB, CHU de Nice l'INRIA, et l'université de Sophia-Antipolis, ce projet a vu le jour en novembre 2005 [42]. Il a bénéficié du savoir-faire du CSTB dans le domaine de la construction des maisons intelligentes des connaissances de l'INRIA en intelligence ambiante et de l'étude approfondie que mène le CHU de Nice dans le domaine de la santé.

Cet appartement a été mis au point pour permettre de mener des expérimentations dans la thématique du maintien à domicile (afin de leurs redonner une certaine autonomie) des personnes atteintes d'handicap ou bien de personnes âgées nécessitant un placement dans des institutions spécialisées.

Les projets menés dans cet appartement (Figure 1.7) touchent plusieurs domaines qui sont : la santé, l'informatique, l'électronique. Plusieurs travaux dans le domaine de l'informatique ont vus le jour, on peut citer ceux de N. Zouba [43][44][45] sur les capteurs de reconnaissance de chute et d'incident humains. En effet le bâtiment intègre une multitude de capteurs, tel que les capteurs de présence. En cas de chute de la personne le système avertit, instantanément l'administrateur qui peut intervenir à n'importe quel moment.

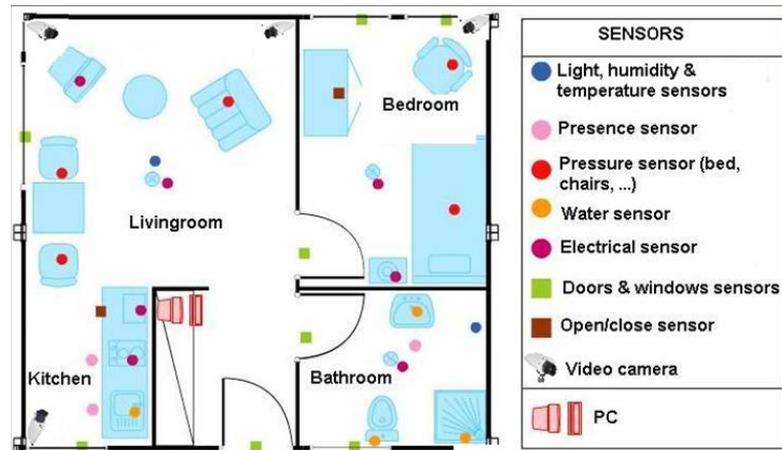


FIGURE 1.7: Architecture et infrastructure réseau de l'appartement GERHOME

L'appartement DOMUS L'appartement DOMUS a été mis en place par l'équipe Multicom du Laboratoire d'Informatique de Grenoble (LIG) [46]. D'une surface de $34m^2$, cet appartement dispose de plusieurs capteurs et actionneurs (luminosité, O/F des fenêtres, température...). L'appartement est doté de 150 capteurs et actionneurs. Le but principal de cet appartement est l'acceptation et l'utilisation des nouvelles technologies. L'appartement est utilisé comme plateforme d'essai des nouvelles technologies et d'évaluation de services interactive dans l'environnement ubiquitaire [47] mise au point par le LIG [48] [49].

L'appartement se compose d'une cuisine/salle à manger, une chambre à coucher, une salle de bains et un bureau. L'appartement est équipé de 7 microphones sans fil, et des caméras fixées au plafond de chaque pièce sauf la salle de bains. La Figure 1.8 décrit l'emplacement exact de la plateforme au sein du LIG. La Figure 1.9 décrit l'architecture de la plateforme expérimentale du LIG (DOMUS).

L'appartement IRoom L'appartement *IRoom* est un projet en partenariat entre LIMSI-CNRS (associé à l'université de Paris-Sud et l'UPMC), Digiteo, SUPELEC, et l'Ile de France. Le but du projet est de mettre en place un appartement doté de plusieurs technologies (capteurs, actionneurs, ...), qui servira par la suite comme une plateforme expérimentale pour les projets portant sur l'informatique ambiante (Bâtiment Intelligent).

L'appartement dispose d'une pièce type séjour qui comprend un mobilier classique et doté de capteurs et d'activateurs (Luminosité, RFID, humidité,...) Figure 1.10.

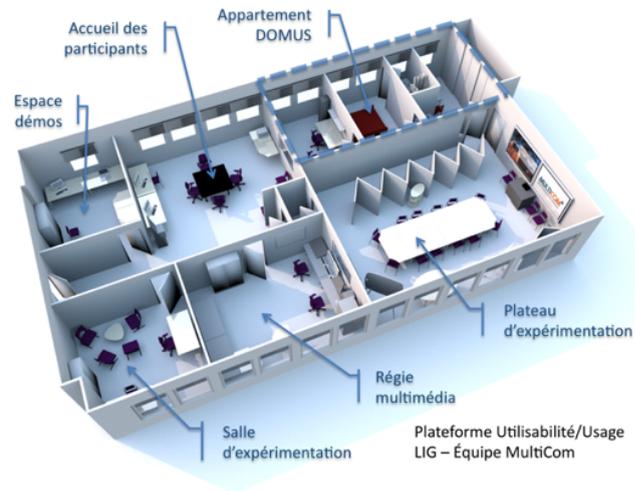


FIGURE 1.8: Plan général du laboratoire



FIGURE 1.9: Plateforme Domus

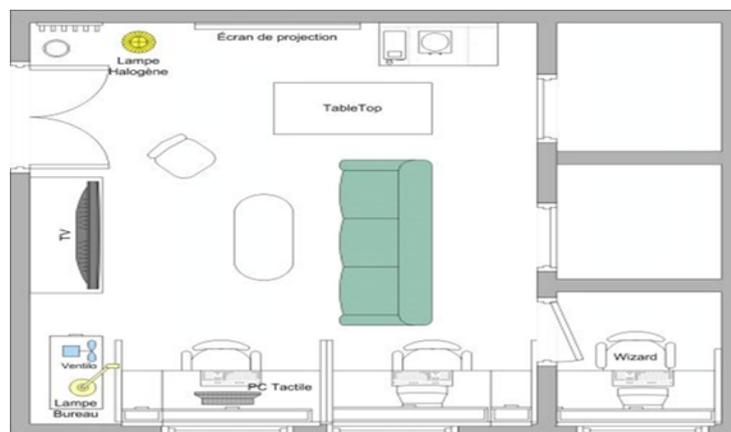


FIGURE 1.10: Vue générale de l'appartement IRoom

1.2.1 Synthèse sur le bâtiment intelligent

Au cours des parties précédentes, nous avons vu l'évolution des systèmes informatiques et leurs intégrations dans notre vie quotidienne. L'introduction de l'intelligence ambiante, qui résulte d'une évolution des dispositifs informatiques a commencé à changer peu à peu l'environnement quotidien des utilisateurs, dont celui de l'habitant. On a vu aussi, que les principaux buts de cette discipline sont d'augmenter le nombre de composants numériques au sein d'un environnement, d'exploiter ces dispositifs pour l'interaction entre l'utilisateur et la machine, et de rendre cette intelligence exploitable en optimisant les services rendus aux utilisateurs.

Nous avons exposé quelques travaux réalisés dans le paradigme d'habitat intelligent. Grand nombre de ces travaux, surtout les projet universitaires, se sont distingués et représentent une activité importante. Le concept de *Living-Labs* illustre l'importance du rôle "d'habitant-concepteur" dans la thématique de l'habitat intelligent.

Afin de permettre au grand public d'adopter le concept de bâtiment intelligent, le coût d'installation est un paramètre qu'il ne faut pas négliger. Aussi, nous pouvons conclure que la notion de réseaux de capteurs est primordiale pour la réussite du bâtiment. Pour cela, nous exposerons dans les parties suivantes, en détail, les réseaux de capteurs et leurs avancées technologiques, ainsi que leurs différents domaines d'applications, notamment pour le bâtiment intelligent.

1.3 Infrastructure réseau du bâtiment intelligent

les Réseaux de Capteurs Sans-Fil (RCSF) sont généralement constitués d'un ou de plusieurs noeuds puits (ou stations de base) et peut-être des dizaines (allant jusqu'à des milliers) de noeuds capteurs répartis dans un espace physique [50]. Le réseau de capteurs sans-fil est souvent caractérisé par sa capacité à surveiller des phénomènes physiques (séismes, incendie, chaleur,...). Les capteurs sont généralement dispersés dans un espace à surveiller appelé *Champs de Captage ou Région d'Intérêt (RI)* [51].

Un RCSF est souvent composé d'un grand nombre de noeuds capteurs dispersés dans la région d'intérêt. Les noeuds collectent périodiquement des données sur le phénomène surveillé et envoient des rapports à l'administrateur du système.

L'architecture d'un RCSF est souvent composée de trois parties : capteurs sans-fil, routeurs (noeud puits), et serveur, comme le montre la Figure 1.11. Les différents noeuds du réseau se relaient l'information jusqu'à atteindre les noeuds routeurs (puits ou sink) qui font office de passerelle entre les noeuds capteurs et le serveur.

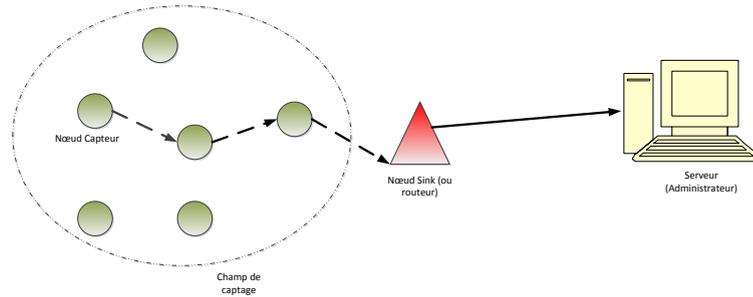


FIGURE 1.11: Architecture d'un réseau de capteurs sans-fil

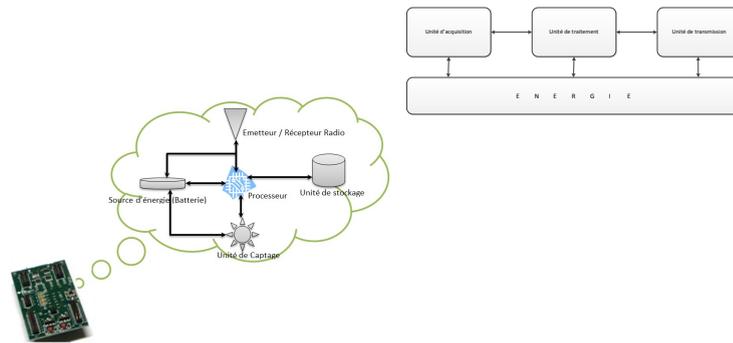


FIGURE 1.12: Anatomie d'un nœud capteur

1.3.1 Nœud capteur

L'entité la plus importante d'un RCSF est le nœud capteur. Un nœud capteur est souvent composé de quatre parties importantes qui sont :

- Unité de traitement
- Unité de captage
- Unité de transmission
- Source d'énergie (généralement des batteries)

La Figure 1.12 montre les différentes parties qui composent un nœud capteur. Dû à leurs tailles très petites et leurs applications qui nécessitent souvent un déploiement dans des environnements limités en termes de ressources, les nœuds capteurs sont généralement alimentés par des piles (utilisées une seule fois et remplacées par la suite) ou des batteries (rechargeables).

1.3.2 Technologie de communication des RCSF

Il existe plusieurs standards de communications qui peuvent être appliqués aux réseaux de capteurs sans-fil on peut citer quatre d'entre elles : Bluetooth[1], WiFi, Wibree[2], UWB[3], et Zigbee[4].

Bluetooth-IEEE 802.15.1[1] Bluetooth est une spécification de l'industrie des télécommunications. Elle utilise une technique radio courte distance destinée à simplifier les connexions entre les appareils électroniques. Bluetooth a été proposé pour transmettre la voix et les données. Il propose une topologie composée de maîtres et d'esclaves (Piconet avec une méthode d'accès de type "TDMA : Time Division Multiple Access"). Il y a seulement sept esclaves actifs par Piconet (255 en mode "parc") et 10 Piconets maximum connectés par Scatternet. Cette technologie est peu utilisée dans les réseaux de capteurs sans fil du fait de la consommation élevée d'énergie, du coût élevé de synchronisation et de la topologie complexe du réseau (le concept Scatternet n'a jamais été mis en œuvre).

Wibree[2] Wibree, aujourd'hui connu sous Bluetooth Low Energy (BLE), est une technique de transmission sans fil créée par Nokia sous forme d'un standard ouvert basé sur Bluetooth qu'il complète sans la remplacer.

Comparé à Bluetooth, Wibree permet un débit du même ordre de grandeur (1 Mb/s) pour une consommation d'énergie 10 fois moindre. Cela permettra d'intégrer cette technique dans de nouveaux types d'équipements : montres, appareils de surveillance médicale ou capteurs pour sportifs. Il prend aussi en charge une topologie en étoile avec un maître et sept esclaves. Sa limite principale est la faible portée de communication : 5-10 m.

UWB-IEEE 802.15.3 [3] Le nouveau standard Wireless IEEE 802.15.3 est un complément très apprécié des utilisateurs friands de morceaux musicaux ou vidéo en streaming (consultation en ligne). Cette norme vient en effet pallier les insuffisances de partage de la bande passante des précédentes normes. Ce protocole peut en effet admettre jusqu'à 245 connections simultanées et procure un débit de 55 Mbps (contre 1 Mbps pour les autres technologies) pour une distance de connexion de 100 mètres environ.

Diffusée sur la fréquence 2,4 GHz, cette norme semble d'autant plus performante qu'elle

garantit l'absence d'interférences avec les autres types de réseaux, ce qui lui permettra de coexister avec les normes de type Wi-Fi (802.11x), 802.15x et Bluetooth.

La norme IEEE 802.15.3 offre de plus une qualité de service accrue, puisqu'il inclut le protocole TDMA (Time Division Multiple Access). Ce dernier permet de gérer les connexions simultanées en fonction de la bande passante disponible afin d'optimiser les transferts et d'éviter les encombrements de réseau. Côté sécurité (le principal point faible des réseaux sans-fil), le nouveau standard bénéficiera là aussi de performances améliorées. Les réseaux IEEE 802.15.3 inclueront en effet l'algorithme de codage AES 128 (Advanced Encryption Data) qui remplace l'ancien protocole DES (Data Encryption Standard). Bien plus performant, il s'appuie sur un système de clés de 256 bits et garantit une meilleure protection des données transférées.

Wi-Fi Wi-Fi est un ensemble de protocoles de communication sans fil régis par les normes IEEE 802.11. Grâce aux normes Wi-Fi, il est possible de créer des réseaux locaux sans fil à haut débit.

Récemment, une nouvelle version de Wi-Fi nommée Wi-Fi à faible puissance "Wi-Fi low power" est proposée pour les réseaux de capteurs. Elle permet de surmonter certains inconvénients de Wi-Fi en termes de consommation d'énergie. Cette nouvelle version est adoptée par certaines entreprises comme : Aginova, Sensicast, STMicro-electronics, GainSpan, Aprion, MicoStrain et Nivis.

Zigbee-IEEE 802.15.4[4] Zigbee est orienté pour être utilisé dans les communications à très faible puissance et sur des distances réduites. Cette technologie, parmi d'autres, est utilisée dans les réseaux de capteurs sans fil pour sa faible consommation énergétique, et sa capacité d'intégrer plusieurs nœuds dans le réseau. Aussi, cette technologie permet une meilleure adaptation du réseau grâce aux différentes topologies qu'elle offre. Sa limite principale est le faible débit notamment lors d'applications multimédia. L'alliance ZigBee formée par un consortium d'entreprises a proposé un standard propriétaire qui offre des fonctionnalités allant de la couche réseau à la couche applicative. Le standard ZigBee repose sur les couches basses du standard IEEE 802.15.4. Dans Zigbee, ce que l'on a appelé précédemment PAN Coordinateur, Coordinateur et nœuds terminaux vont s'appeler respectivement le ZigBee Coordinateur (ZC), ZigBee Router (ZR) et ZigBee End Device (ZED) qui ne peut communiquer qu'avec son coordinateur (ZR). Zigbee ou plus précisément la couche réseau du standard Zigbee a comme tâches les opérations suivantes :

- Découverte du voisinage

Standard	Bluetooth	UWB	Zigbee	Wi-Fi
IEEE spec.	802.15.1	802.15.3	802.15.4	802.11
Frequency band	2.4 GHz	3.1-10.6 GHz	868/915 MHz; 2.4 GHz	2.4 GHz; 5 GHz
Max signal rate	1 Mb/s	110 Mb/s	256 Kb/s	54 Mb/s
Nominal range	10 m	10 m	10-100 m	100 m
Nominal TXpower	0-10 dBm	-41.3 dBm/MHz	(-25) dBm	-0 15-20 dBm
Number of RF channels	79	1-15	1/10; 16	14 (2.4 GHz)
Channel bandwidth	1 MHz	500 MHz-7.5 GHz	0.3/0.6 MHz; 2 MHz	22 MHz
Modulation type	GFSK	BPSK, QPSK	BPSK (+ASK), O-QPSK	BPSK, QPSK, COFDM, CCK, M-QAM
Spreading	FHSS	DS-UWB, MB-OFDM	DSSS	DSSS, CCK, OFDM
Coexistence mechanism	Adaptative freq. hopping	Adaptative hopping	Dynamic freq. selection	Dynamic freq. selection transmit power control (802.11h)
Basic cell	Piconet	Piconet	Star	BSS
Extension of the basic cell	Scatternet	Pèer-peer	Cluster tree-mesh	ESS
Max number of cell nodes	8	8	>65000	2007
Data protection	16-bit CRC	32-bit CRC	16-bit CRC	32-bit CRC

TABLE 1.1: Tableau comparatif entre les différents standards de communications [18]

- Création de la topologie
- Adressage
- Routage

La couche application de ZigBee propose différents profils applicatifs selon l'utilisation désirée du réseau. Nous citerons par exemple :

- Domotique : éclairage, système d'alarme, chauffage ...
- Zigbee Smart Energy
- Santé : suivi et surveillance de patients
- Contrôle industriel : englobant la domotique et le contrôle de production
- Zigbee Telecommunication Applications

Le tableau Tab 2.2 présente une vue comparative entre différentes technologies de communications les plus utilisées dans les RCSFs.

1.3.3 Domaines d'applications des Réseaux de Capteurs Sans-Fil

Vu leurs petites tailles et leurs capacités de surveillance d'un bon nombre de phénomènes physiques, les réseaux de capteurs sans fil ont connu un grand succès dans nombreux

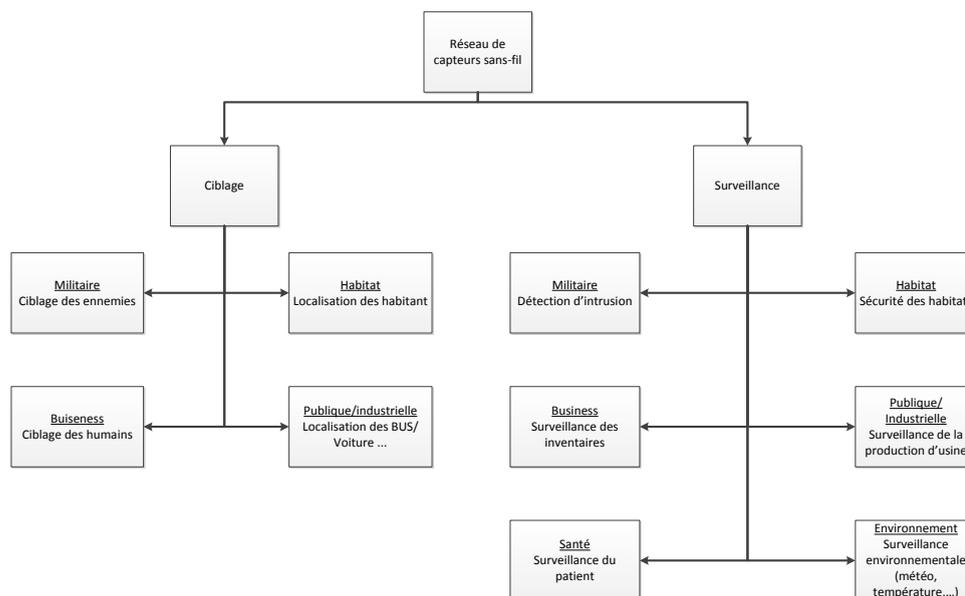


FIGURE 1.13: Panorama des applications d'un RCSF

domaines tels que : la santé, le militaire, l'environnement et le bâtiment intelligent (voir la Figure 1.13).

Le Militaire Les RCSF constituent une part intégrale dans le domaine militaire surtout les applications de surveillance, reconnaissance et ciblage. L'aspect *ad-hoc* des RCSFs offre plusieurs avantages tels que : le déploiement rapide et l'auto-organisation qui les rendent très utilisés dans les applications militaires [52].

Les applications les plus importantes des RCSFs dans le militaire sont :

- La surveillance des champs de batailles
- La surveillance des munitions et équipements
- Reconnaissance des forces opposantes
- Ciblage
- Détection des attaques chimiques

L'environnement Les RCSFs ont aussi investi le domaine de l'écologie et l'environnement. Diverses, sont les applications d'environnement qui utilisent la technologie des capteurs sans fil. On peut citer par exemple, la surveillance des oiseaux migrateurs ainsi que les animaux et les insectes, la surveillance des récoltes et du bétail, les systèmes d'irrigation intelligents, la détection des incidents naturels (incendies et inondations) et l'étude de l'empreinte carbone [53].

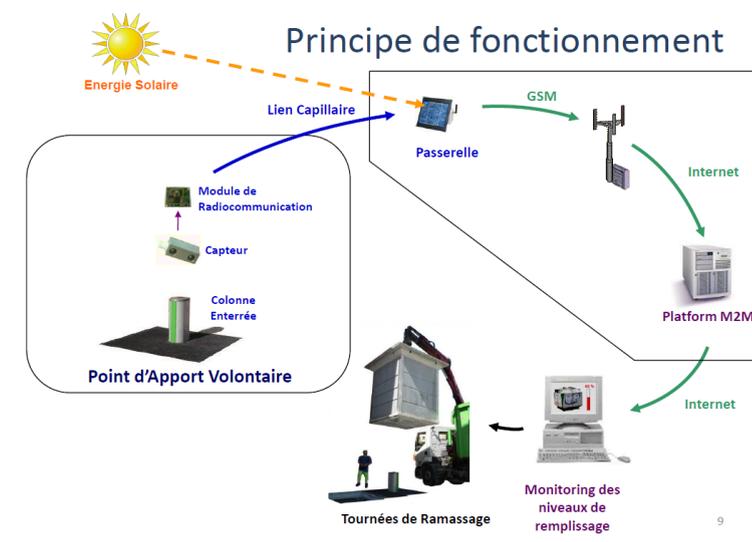


FIGURE 1.14: Principe de fonctionnement de la collecte des verres

Prévention contre les catastrophes naturelles Les travaux d'A. Lellah et al. [54] qui se sont déroulés au G-SCOP (université de Grenoble), où l'équipe de recherche a développé un système de collecte de verres recyclables totalement automatisé. Des capteurs sans fil sont placés sur des colonnes enterrées sous le sol et collectent périodiquement des informations sur le remplissage de ces dernières. Si une colonne est remplie, le capteur embarqué envoie une demande à l'administrateur du système, qui informe instantanément les services compétents afin de collecter les verres et de vider cette colonne Figure 1.14.

On peut citer aussi les travaux de B. Son et al. [55], de l'université de Daego en Corée, qui portent sur la détection des incendies des forêts. Les chercheurs ont mis au point un système de détection et gestion des incendies des forêts. En cas d'incendie les capteurs, éparpillés sur un champ (forêt), envoient un signal d'alarme. Dès que l'incendie est notifié, les autorités compétentes se déplacent sur le lieu de l'incident grâce aux données de localisations des capteurs.

La santé Une autre application des RCSFs est le suivi des patients atteints d'une maladie ou d'un handicap. Vu leurs capacités de surveillance de phénomènes, de détection et de localisation, les capteurs sans fil sont utilisés dans le domaine de la télémédecine et d'aide aux patients.

On peut citer les travaux de C. Jacquet et al. [56] qui ont proposé un système d'aide au déplacement des non-voyants. Pour ce faire, un système est intégré dans la canne du patient, à l'aide de capteurs (infrarouge, ultra-sons ...) le système guide le non-voyant

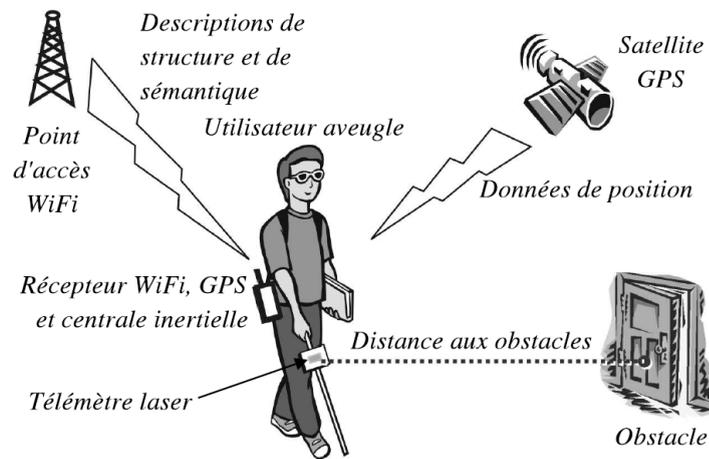


FIGURE 1.15: Système d'aide aux non-voyants

durant tout son trajet en l'informant des différents obstacles rencontrés durant son trajet (escalier, porte ...), comme l'explique la Figure 1.15.

Le bâtiment intelligent Pour avoir des informations sur son environnement, ainsi que sur les phénomènes physiques qui se produisent dans l'enceinte du bâtiment, ce dernier nécessite l'implémentation d'une technologie dotée d'une capacité de perception de l'environnement (ex. : RCSF). Vu leur coût et leur tailles qui ne cessent de diminuer, les capteurs sans fil sont généralement favorisés et utilisés comme infrastructure réseau dans les habitats intelligents (voir section 1.2).

1.3.4 Limites des réseaux de capteurs sans-fil

Selon les travaux menés dans [51, 57–59], les RCSF souffrent de plusieurs inconvénients (limites), nous pouvons les classer comme suit :

- *Durée de vie* : En effet, les RCSF sont souvent alimentés par une source d'énergie restreinte telle que les piles ou les batteries ce qui rend leurs durées de vies limitées.
- *Mémoire* : vu leur taille qui approche parfois les 1cm^3 , le coût d'intégration d'une grande mémoire augmente fortement. Aussi, il est difficile d'intégrer toutes les parties qui composent ce-dernier en un seul boîtier aussi petit. Les capteurs sans fil disposent d'une mémoire très petite et ne peuvent stocker qu'une partie des informations collectées.
- *Calcul* : les capteurs sont dotés aussi d'un processeur mais ne peuvent pas effectuer des opérations très grandes.

- *Communication* : Aussi les communications au sein du réseau peuvent être influencées par les différents obstacles dans le terrain d'intérêt, et cela, comme toutes les communications sans fil.

1.4 Outil d'aide au déploiement d'un RCSF

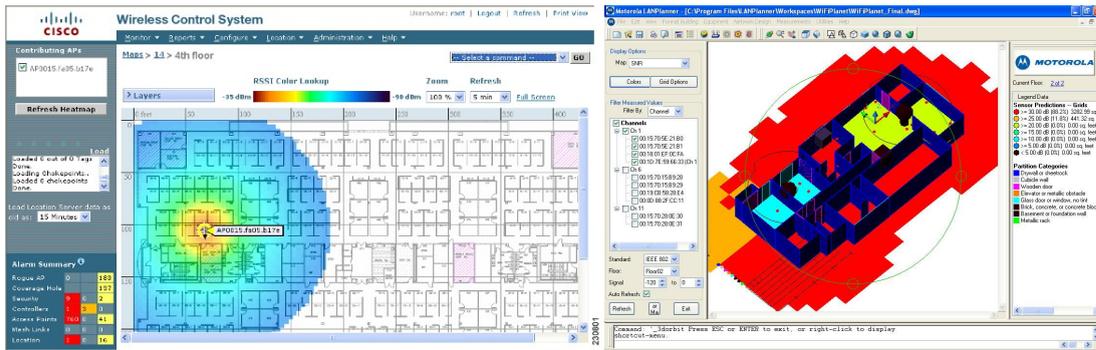
Selon [25] les RCSF sont la solution la plus utilisée dans le domaine du bâtiment intelligent. Les auteurs listent aussi les problèmes majeurs rencontrés lors du déploiement de ce type de réseau dans le bâtiment. Ces problèmes sont comme suit : une durée de vie limitée, problèmes de connectivité et de couverture, et un coût de déploiement et de maintenance élevé notamment dans des emplacements inaccessibles.

Les travaux menés dans [29] et dans [25] montrent qu'une bonne conception du réseau (nombre de noeuds, placements des noeuds, topologie, ...) est essentielle pour avoir un réseau viable et rentable. Généralement le placement des noeuds constituant le réseau est réalisé en se basant sur une approche de prédiction ou un concepteur de réseau désigne, selon des connaissances dans le domaine, des placements ou les noeuds peuvent être installés. Cette méthode peut mettre en œuvre un réseau dans des délais très courts mais elle nécessite toujours la présence d'une personne (technicien ou autre) formé dans le domaine de d'installation des RCSF. Aussi, cette méthode peut être fiable dans des réseaux à petite échelle (petit nombre de noeuds), mais on voit bien ces inconvénients dans une mise à grande échelle (plusieurs étages, plusieurs noeuds capteurs ...).

Pour remédier à ces inconvénients, des solutions automatisées ont vu le jour. Plusieurs logiciels et outils de planification de réseaux locaux et de placements optimaux des stations de bases et routeurs ont été proposés par des sociétés commerciales. On peut citer quelques exemples d'outils : AirMagnet Planner[60](Figure 1.16c), LANPlanner (Motorola)[61](Figure 2.13d), Cisco Prime Infrastructure (CISCO systems)[62] (Figure 2.13c), CINDOOR, GISAR [63], qui sont des outils destinés à la conception de réseaux WLAN. Leurs interfaces graphiques proposent les placements de point d'accès d'un réseau sans-fil type WLAN (Wireless Local Area Network).

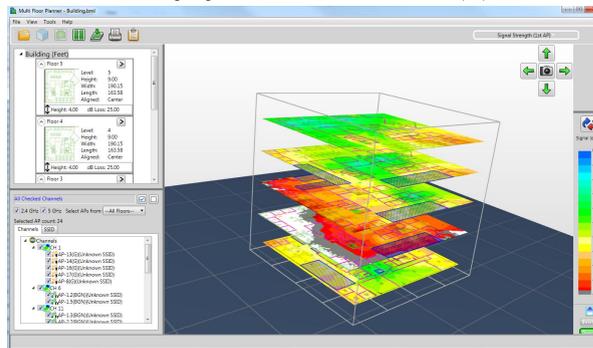
1.4.1 Architecture de l'outil de déploiement des RCSF

L'outil d'aide au placement (où outil de déploiement) a été défini par A. Mc Gibney et al. [30] comme étant un système automatique capable d'assister les concepteurs durant la phase de déploiement du réseau de capteurs sans fil, et cela jusqu'à sa mise en marche.



(A) Cisco Prime Infrastructure [62]

(B) LANPlanner [61]



(C) AirMagnet Planner [60]

FIGURE 1.16: Différents outils d'aide au déploiement de réseau LAN

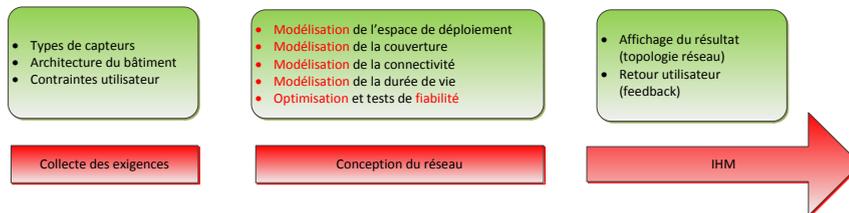


FIGURE 1.17: Différentes phases de l'outil d'aide au placement

Selon l'auteur, l'outil doit comporter trois parties distinctes qui se succèdent l'une à l'autre : (1) Collecte des besoins utilisateur, (2) Conception du réseau et finalement, (3) Affichage des résultats. La Figure 1.17 représente le processus de déploiement.

Collecte des exigences Pour pouvoir procéder à la conception du réseau ainsi que le calcul des positions optimales des nœuds capteurs il est nécessaire de procurer des données sur l'environnement où va être déployé le réseau. Ces informations peuvent être classées selon leur nature : architecture du bâtiment, propriété des nœuds, contraintes de placement.

- *Architecture du bâtiment* On définit dans cette partie tous les obstacles pouvant réduire ou atténuer les ondes radios (mur, étage, porte, ...) et qui peuvent causer des dysfonctionnement du RCSF.
- *Caractéristiques des nœuds* On définit dans cette phase les différentes caractéristiques du nœud capteur/routeur : propriété de l'antenne, capacité de la batterie, portée de l'antenne, modèle de propagation utilisée,...
- *Contraintes utilisateur* Durant cette phase l'utilisateur de l'outil doit définir tous les équipements présents dans les pièces du bâtiment et qui peuvent avoir une influence sur la communication ou la collecte des données. Par exemple, un capteur de température indiquera des informations erronées s'il est placé à côté d'une source de chaleur. Pour cela, l'utilisateur devra fournir toute la liste des équipements (électroménagers, système de ventilation et de chauffage, ...).

Conception du réseau Dans cette partie, non visible par l'utilisateur, le système va définir le nombre nécessaire de nœuds capteurs, ainsi que leurs emplacements optimaux. Avant de procéder à cela, une bonne modélisation du système est essentielle. La phase de conception est divisée en deux sous-parties : Modélisation du système et Optimisation. Ces deux parties vont être abordées dans les prochains chapitres.

Dans la partie modélisation, le système va traduire les requêtes, qui ont été saisies par l'utilisateur dans la phase de collecte de données, en un modèle mathématique pouvant être exploité par la partie optimisation. Après cela, le système va définir des positionnements optimaux, selon des métriques qu'il calcule, grâce à des méthodes d'optimisation prédéfinie.

Affichage des résultats L'outil devra disposer d'une interface graphique simple grâce à laquelle il recueillera toutes les données nécessaires au déploiement telles que : plan du bâtiment, contraintes de placement. L'autre fonction de cette interface est l'affichage des résultats obtenus.

1.5 Conclusion

Dans cette partie, nous avons décrit le contexte et la problématique de la thèse. Nous avons défini l'habitat intelligent et identifié les infrastructures réseaux nécessaires pour son bon fonctionnement. Nous confirmons que le réseau de capteurs sans fil est la technologie la plus adaptée aux bâtiments intelligents.

Au cours de cette partie, nous avons abordé brièvement la notion de RCSF et les différentes parties qui le composent. Nous avons présenté les différentes applications

qui se basent sur cette technologie. Enfin, nous avons exposé les limites de cette technologie, qui reste malgré son avancée, un terrain riche en terme de recherches scientifiques. Nous allons nous focaliser, dans les prochaines parties, sur quatre critères essentiels, à optimiser, qui sont :

- *Le coût d’installation des infrastructure* : ce critère est essentiel pour pouvoir atteindre un maximum de public utilisant la technologie des habitats intelligents. Malheureusement, comme on l’a vu avec les deux projets du Pr. Sakamura, le coût d’installation des habitats intelligents reste trop élevé ce qui limite son utilisation à des laboratoires de recherches et à certaines personnes pouvant accéder à cette dernière.
- *Durée de vie du réseau* : en effet, la durée de vie du réseau est limitée par l’épuisement des ressources d’énergie embarqué dans chaque nœud capteur. Nous allons détailler cette dernière, et nous allons essayer de proposer quelques méthodes d’optimisation en vue de la maximiser.
- *Connectivité (Communication)* : La communication entre capteurs dans le réseau est primordial, un bon fonctionnement du réseau induit au bon fonctionnement du bâtiment et *vice-versa*.
- *Couverture des évènements* : La tâche principale d’un RCSF est la collecte de données sur des phénomènes physiques. Pour cela, il faut garantir une excellente couverture de tous les endroits du bâtiment.

Nous allons aborder dans le prochain chapitre les différents types de modélisation du problème de déploiement des RCSFs. Nous exposerons par la suite les modèles retenus qui nous seront utiles lors de la phase d’optimisation du déploiement.

Chapitre 2

Modélisation de la problématique et méthodes de résolution

2.1 Introduction

Les performances d'un RCSF dépendent principalement du processus de déploiement[64]. Le processus de déploiement définit le nombre et les positions optimales des différentes composantes qui forment le réseau. Les différentes métriques de performances : couverture, durée de vie, connectivité, dépendent essentiellement du positionnement des nœuds ainsi que de la topologie du réseau adoptée. Il existe toutefois quelques travaux sur le problème de déploiement des RCSF, mais qui ne peuvent être appliqués aux environnements *indoor*.

Dans le chapitre précédent, nous avons défini l'architecture et les composantes principales d'un outil de déploiement. Nous avons remarqué, que la conception d'un RCSF nécessite une modélisation des différents paramètres tels que : connectivité, couverture, durée de vie et coût. L'outil de déploiement des RCSF devra prédire les différentes positions des nœuds formant le réseau en se basant sur les données entrées par l'utilisateur qui se présentent comme suit :

- Plan du bâtiment (espace de déploiement) : les obstacles tels que : les murs, les fenêtres, . . . , seront définis dans cette phase.
- Paramètres des capteurs/routeurs : dans cette partie nous allons définir les principaux paramètres des nœuds à savoir : la capacité de la batterie embarquée, la puissance de transmission du signal, le type de nœud (routeur ou capteur), . . .
- Équipements présents dans le bâtiment : l'utilisateur doit définir les emplacements et le type d'équipements présents dans le bâtiment qui peuvent avoir un impact sur le

bon fonctionnement du réseau (ex. : les sources de chaleurs, climatisation, chauffage, ...)

Une fois ces données précisées, on procède à l'adaptation des modèles proposés. Après la phase de modélisation du réseau, nous procédons à l'optimisation du placement des nœuds. Durant cette phase, l'outil se basera sur les modèles adaptés pour prédire les emplacements des différents nœuds formant le RCSF. Finalement, l'outil disposera d'une interface graphique pour visualiser la topologie proposée. Ici le choix est laissé à l'utilisateur qui pourra à chaque moment consulter ou changer la solution proposée. Dans ce chapitre, nous commençons par exposer les différents travaux existants dans la littérature sur la modélisation du comportement d'un RCSF. Puis, nous aborderons successivement les modèles proposés qui seront utilisés dans la phase d'optimisation multi-objectifs (chapitre 3).

2.2 Modélisation des critères de déploiement

Dans cette partie, Nous présentons un état de l'art sur la modélisation des critères de déploiement d'un RCSF. Ces derniers sont comme suit : environnement de déploiement, communication (connectivité), couverture des événements, et durée de vie. Un modèle mathématique est conçu pour chaque critère de déploiement. Ces modèles représentent les données d'entrée de la phase d'optimisation du placement des nœuds. La bonne prise en compte de tous les paramètres liés au déploiement des RCSF permet d'avoir des résultats satisfaisant les besoins des utilisateurs. Les modèles que nous utilisons seront présentés par la suite dans la section 2.3.

2.2.1 Espace de déploiement

Nous considérons un déploiement 2D où l'espace de déploiement est modélisé par une grille où chaque cellule de la grille représente un emplacement potentiel d'un nœud du réseau Figure 2.1. Après la saisie du plan du bâtiment (par l'utilisateur) ce dernier est divisé en un nombre $L * W$ de cellules où L représente la longueur du bâtiment et W sa largeur. Chaque cellule a une aire de $1m^2$, les nœuds du réseau peuvent être placés dans le barycentre des cellules.

2.2.2 Connectivité

L'un des critères essentiels requis lors du déploiement d'un réseau de capteurs sans fil est la connectivité [65]. Le réseau doit assurer une très bonne communication entre ses

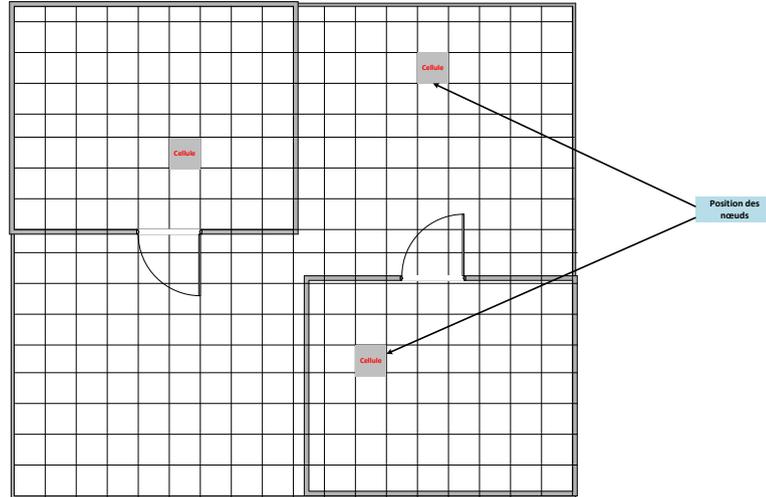


FIGURE 2.1: Plan du bâtiment discrétisé en cellule

différents nœuds. Chaque nœud capteur/routeur doit pouvoir communiquer au minimum avec un autre nœud et cela en vue d'avoir les informations nécessaires sur les phénomènes à surveiller.

La connectivité du réseau est généralement assimilée à la propagation d'ondes radio. Cette dernière est une modélisation du comportement du signal émis par un nœud du réseau. On peut dire qu'un nœud est connecté à un autre (peut transmettre les informations), si et seulement si, chaque nœud peut transmettre/recevoir des données de l'autre. Cela peut être défini par la puissance de réception calculée au niveau d'un nœud par rapport à un autre.

Un RCSF est généralement représenté par un graphe $G = (V, E)$, où V représente l'ensemble des nœuds et E l'ensemble des liens associés. Si un lien $(N_e, N_r) \in E$ existe, cela veut dire que N_e, N_r sont des voisins et peuvent communiquer (envoyer/recevoir des informations).

L'ensemble des voisins d'un nœud quelconque N_i est exprimé comme suit :

$$\eta(i) = \{N_j / (N_j, N_i) \in E \vee (N_i, N_j) \in E\} \quad (2.1)$$

Les liens sans-fil sont déterminés selon la valeur de la puissance du signal reçu par un nœud (RSSI : Received Signal Strength Indicator). Ainsi, chaque lien peut être défini comme suit :

$$E = \{(N_e, N_r) \in V^2 \mid e \neq r \wedge RSSI(er) \geq RX_{Tsh}\} \quad (2.2)$$

Où $RSSI(er)$ représente la puissance du signal transmis par le nœud N_e et calculée au niveau du nœud N_r , RX_{Tsh} est la sensibilité de l'antenne de réception (elle représente la puissance minimale autorisée pour établir une connexion). Il existe une multitude

de modèles de propagations radio, qui peuvent exprimer le comportement de l'onde transmise par un nœud dans un environnement précis. Dans les parties suivantes, nous exposons les modèles les plus importants.

2.2.2.1 Étude de l'existant

Tout a commencé avec l'apparition des équations de Maxwell en 1873[66], qui mettent en relation les deux domaines de l'électricité et du magnétisme qui ne formeront qu'un à l'avenir "l'électromagnétisme". En 1886, le Prof. Heinrich Rudolph Hertz a découvert le premier système électromagnétique. Il a fallu attendre l'année 1901 pour avoir la première communication sans fil, c'était grâce à Guglielmo Marconi qui a pu, cette année là, transmettre le 1^{er} signal transatlantique de Podlhu à Cornwall, jusqu'à St. John à Newfoundland [67]. Durant la seconde guerre mondiale, des nouvelles technologies de transmission sans fil sont apparues (nouvelle génération d'antennes de transmissions) grâce à l'introduction de nouveaux appareils tel que le *magnétron* qui est une source d'onde avec des fréquences de 1 GHz et plus. De nos jours, des modèles miniatures tels que les antennes de téléphones portables sont apparus qui offrent plus de possibilités d'innovation.

Allant du modèle le plus simple au plus complexe, nous présentons un résumé de l'évolution des modèles de propagation. Balanis [68], offre dans son livre une revue littéraire de la théorie des antennes et de son évolution. On peut retrouver le modèle de propagation en espace-libre proposé par FRIIS qui est exposé en détails. De nos jours, on utilise de plus en plus d'appareils communicants. La nécessité de modèles de propagation plus adaptés aux environnements intérieurs devient de plus en plus importante. Pour cela, plusieurs modèles de propagations en environnements intérieurs (ex. bâtiments) ont vus le jour. Dans [69], un modèle basé sur la théorie du lancer de rayons est présenté. Dans ce modèle l'onde électromagnétique est représentée par des rayons qui sont générés par un transmetteur et lancés dans un espace 3-D par la suite. La technique adoptée dans ce modèle prend en considération les différents phénomènes liés à la réflexion de l'onde sur les murs et le sol. Dans [14], les auteurs ont présenté une étude des différents facteurs d'atténuations de l'onde liés aux murs construits en béton ou en matériaux composites. Cet article pourrait servir d'une référence pour des travaux (modèles) nécessitant des données sur les facteurs d'atténuations. Aussi, ce travail pourrait être considéré comme étant une base pour tous les modèles de propagations en environnements intérieurs. C. F. Yang et al [70], présentent dans leur article une modélisation de la propagation d'une onde radio dans un bâtiment. Cette modélisation se base, elle aussi, sur la méthode de lancer de rayons afin de prédire la puissance d'un

signal reçu. Une étude comparative entre les résultats de simulation du modèle proposé et ceux issus de mesures effectuées dans le bâtiment NTUST (Université nationale des sciences et des technologies de Taiwan) a été menée. Dans [71], un nouveau modèle de propagation dans un environnement intérieur a été proposé. L'étude a été menée au sein d'un bâtiment de l'université Jiao Tong à Shanghai. Le bâtiment est composé de plusieurs salles de cours où les murs d'enceinte sont construits en béton et les séparations en bois. Une étude comparative entre les résultats de simulation et expérimentaux ont été menée. Les résultats montrent que le modèle proposé est efficace, car il n'y a pas de grand écart entre les mesures effectuées et les résultats issus de la simulation. Dans [72], A. E. Saikh et al présentent dans leur papier un nouveau modèle de propagation radio déterministe qui se base sur la théorie de la géométrie optique et la méthode de lancer de rayons. Certes, ces méthodes sont coûteuses en termes de temps de calcul et de mémoire, mais restent des méthodes efficaces pour déterminer la propagation radio dans un environnement intérieur. Les auteurs ont ensuite fait une étude comparative entre les résultats issus de la simulation et ceux mesurés. Après validation de leur méthode, les auteurs ont proposé un outil doté d'une interface graphique capable de cartographier la propagation d'onde d'une (ou plusieurs) antennes de transmission (s) positionné (es) par l'utilisateur grâce à la GUI.

L'un des problèmes majeurs rencontrés avec ce type de modèle est leur temps de calcul très grand. Le modèle de lancer de rayons est certes efficace, car il prends en compte tous les phénomènes de : réflexion, réfraction, diffraction. . . , mais consomme trop de temps de calcul. Pour cela, de nouvelles techniques ont été proposées. Dans [73], M. Lott et I. For- kel proposent un modèle plus simple à implémenter et adapté aux bâtiments. Ce modèle sus-cité MWF (Multi-Wall-Floor) se basant sur des anciens travaux tels que le modèle de FRIIS et LOS (Line-Of-Sight). En effet, ce modèle ne prends en compte qu'une seule onde transmise en ligne direct entre l'émetteur et le récepteur. Si cette ligne croise un obstacle (mur, fenêtre, porte. . .) l'onde est atténuée en fonction du facteur d'atténuation de l'obstacle. Sinon, l'onde se propage en suivant le modèle Line-Of-Sight (LOS). B. Sujak et al [74] ont proposé dans leurs article un modèle adapté aux applications indoor nommé MWL (Multi-Wall Linear). Ce modèle est une hybridation entre les anciens modèles : MWC (C : Classic) et MWE (E : Extended), avec le rajout d'une nouvelle fonction linéaire $\alpha(d)$. Si l'onde transmise ne rencontre aucun obstacle, elle se propage suivant la fonction $\alpha(d)$. Sinon, la valeur du facteur d'atténuation de l'obstacle est soustraite de la puissance de l'onde. Dans [75], les auteurs ont effectués plusieurs mesures de RF (Radio Fréquences) dans un bâtiment d'un système Wi-Fi opérant sous 2.4 GHz. Basé sur ces mesures, les auteurs ont entrepris une étude comparative entre plusieurs modèles de propagation et les résultats mesurés. Finalement, les auteurs ont validés le modèle MWF pour les environnements indoor et ont prouvé que les effets de Shadowing suivent une distribution log-normal. Les auteurs ont intégré une fonction qui génère des nombres

suivant une distribution log-normal afin de prendre en compte l'effet Shadowing. Dans [76], une version améliorée du modèle MWF pour des bâtiments construits en béton a été proposée. Au début, une campagne de mesures a été effectuée au sein d'un bâtiment afin de déduire les facteurs d'atténuations des différents obstacles présents dans le bâtiment. Le modèle a été testé par la suite sur des systèmes opérant sous la norme Wi-Fi à 2.4 GHz. L'équation calculant l'atténuation de l'onde a été décomposée en deux parties : (1) propagation en espace-libre si pas d'obstacle, (2) changement du facteur d'atténuation lié à l'environnement et prise en compte des atténuations des obstacles statiques (murs, portes, fenêtres). Le modèle a été ensuite comparé aux mesures effectuées, et validé par expérimentation.

Modèle de FRIIS (espace-libre) L'un des premiers modèles de propagation d'ondes radios proposé dans la littérature est le modèle de Friis communément appelé *équation de transmission de Friis*[6]. Ce modèle met en relation la puissance du signal reçu et celui émis entre deux antennes séparées par une distance d (Fig 2.2). Dans ce modèle, les antennes de transmissions sont considérées comme étant isotropes (rayonnement uniforme dans toutes les directions Fig 2.3)[77]. L'antenne dispose d'un gain noté $G_{t/r}$ (émission/réception respectivement).

Soit deux antennes (une émettrice et l'autre réceptrice), le rapport puissance de réception/transmission selon Friis est donné par l'équation suivante :

$$PL(d) = 10 * n * \log\left[\left(\frac{\lambda}{4\pi R}\right)^2 * G_t * G_r\right] \quad (2.3)$$

Cette équation est connue sous le nom de l'équation de transmission de Friis, elle met en relation la puissance du signal reçu (P_r) par rapport au signal d'entrée de l'antenne de réception (puissance de transmission P_t). Le terme $(\frac{\lambda}{4\pi R})^2$ est appelé facteur de perte en espace libre (en anglais : free-space loss factor) et prend en compte toutes les pertes dues à la propagation sphérique de l'onde. Cette équation de propagation d'onde radio a été la base des autres modèles proposés par la suite.

Modèle Line Of Sight (LOS) Dans une communication LOS (Line-Of-Sight), le chemin principal de l'onde est le chemin direct entre un transmetteur et un récepteur (en prenant en considération la courbure de l'onde). Ce type de communication est principalement utilisée dans les systèmes suivants[78] :

- Communication terrestre fixe point-à-point
- Communication terrestre fixe point-à-multi-points
- Communication radio mobile
- Communication satellite

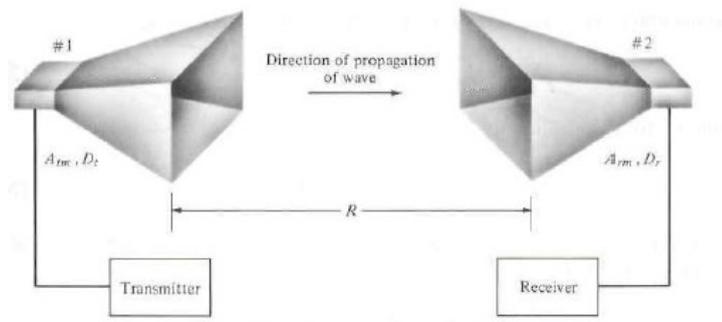
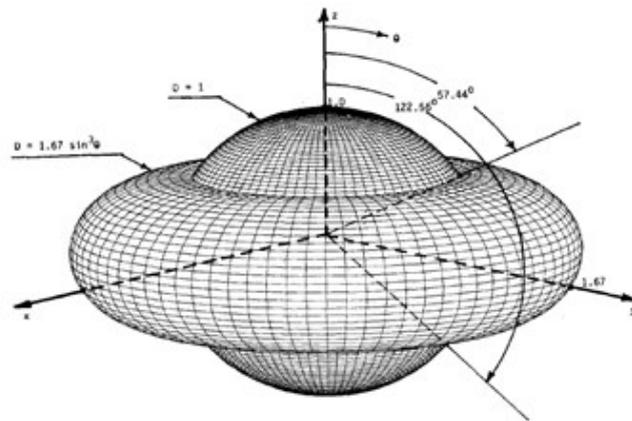
FIGURE 2.2: Deux antennes séparées par une distance R [6]

FIGURE 2.3: Diagramme de rayonnement 3-D en sphère d'une antenne isotrope[6]

- Communication point-à-point fixe utilisant des répéteurs passifs

Les principaux facteurs affectant une propagation LOS sont :

- Atmosphère terrestre
- Topographie terrestre : montagne, mer, désert...
- Construction : bâtiment, ponts ...

Les paramètres essentiels considérés dans un lien LOS typique (Figure 2.4), sont comme suit :

- Niveau du signal reçu à l'antenne réceptrice ($RSSI$) en dBm
- Puissance du signal émis par l'antenne émettrice (P_{Tx}) en dBm
- Niveau de seuil récepteur (puissance minimale du signal pour pouvoir établir une connexion) (P_{Tsh}) en dB
- Perte de signal en espace libre (FSL : Free-Space Loss), en dB
- Pertes liées au Guide d'ondes (Waveguide) : du transmetteur (LFT : Loss in Feeder Transmitter) et récepteur (LFR : Loss in Feeder Receptor) en dB

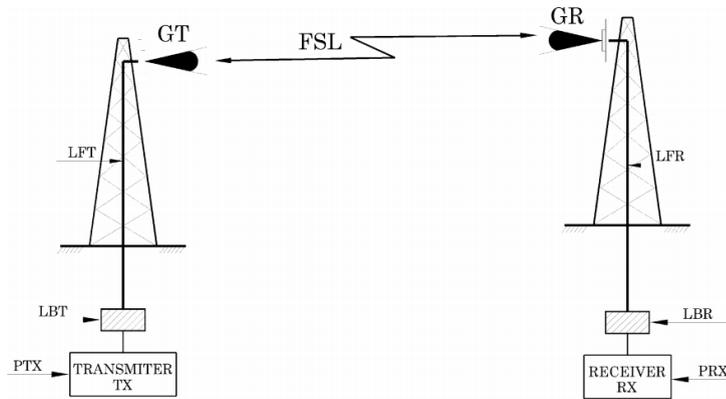


FIGURE 2.4: Lien radio typique LOS[7]

- Pertes dues aux branchements dans l’antenne émettrice et réceptrice (LBT/LBR : Loss in Branching of the Transmitter/Receptor) en dB
- Gain de l’antenne de transmission (GT : Gain of the Transmitter antenna) et de réception (GR : Gain of the Receptor antenna)
- Pertes diverses (LM :Miscellaneous Losses) en dB
- Toutes les pertes de propagation autres que la perte en espace libre (L_t) en dB

En considérant la perte et le facteur de gain dans une liaison radio typique (LOS), le niveau du signal reçu est calculé comme suit :

$$RSL = P_{Tx} + GT + GR - (FSL + LFT + LFR + LBT + LBR + LM + L_t) \quad (2.4)$$

Modèle Multi-Wall (MWM) Mootley et Keenan [79], ont développés un modèle de propagation d’ondes dans un environnement indoor (bâtiment). Ce modèle prend en compte toutes les atténuations dues à la pénétration de l’onde radio dans les différents obstacles présents dans le bâtiment (murs, portes, fenêtres, étage...). Les atténuations varient selon l’épaisseur et le matériel de construction de l’obstacle. Les murs et étages de la même catégorie (même épaisseur et matériaux de construction) ont le même effet (atténuation) sur l’onde radio.

Ce modèle étant basé sur le modèle (LOS), il ne prend en compte que les atténuations de l’onde directe entre l’émetteur et le récepteur. Aucune autre perte n’est considérée : réflexion, diffraction, ... Le modèle peut être représenté par l’équation suivante :

$$PL_{MWF}[dB] = FSL + 10 * n * \log_{10}(d) + \sum_{i=1}^I \sum_{k=1}^{K_{wi}} (L_{wik}) + \sum_{j=1}^J \sum_{k=1}^{K_{fj}} (L_{fjk}) \quad (2.5)$$

$$RSSI[dBm] = P_{TX}[dBm] - PL_{MWF}[dBm] \quad (2.6)$$

Où :

- FSL : Free-Space Losses
- L_{wik} : est l'atténuation du k^{eme} mur de la catégorie i
- L_{fjk} : est l'atténuation du k^{eme} étage de la catégorie j
- K_{wi} : est le nombre des murs de la catégorie i
- K_{fj} : est le nombre d'étages de la catégorie j

Ce modèle est le plus utilisé dans des applications indoor (bâtiment intelligent). Cela est dû à sa facilité d'implémentation et son temps de calcul réduit. Ceci nous permet de simuler plusieurs scénarios (topologies réseaux) au même temps (en parallèles).

Modèle à lancer de rayons Ce modèle a été développé initialement pour la synthèse d'image. Le principe de cette méthode consiste à émettre depuis une source un nombre de rayons (dans toutes les directions possibles) et suivre leurs développements dans un environnement donné. Dans ce modèle l'onde transmise est représentée par des rayons générés par un transmetteur et lancés dans un espace 2D/3D. Chaque rayon est lancé suivant différents angles et son chemin est suivi jusqu'à ce que la puissance de l'onde soit inférieure à un seuil prédéfini[70].

Ce modèle est complet, du point de vue qu'il considère tous les phénomènes que peut subir une onde électromagnétique, à savoir : diffraction, réfraction, ... Par contre, il reste peu utilisé car il nécessite beaucoup de mémoire et un temps de calcul considérable pour arriver à retracer tous les chemins des rayons générés. Ce temps de calcul augmente avec le nombre de transmetteurs, ainsi que la superficie de l'environnement.

2.2.3 Détection des événements par le RCSF

La couverture des événements est l'une des caractéristiques primordiales abordées lors du déploiement des RCSF [80]. Cette caractéristique représente la fonction principale du réseau qui est la surveillance des événements et la collecte des données sur les phénomènes se produisant dans l'espace environnant. La couverture peut être considérée comme un objectif (à maximiser en général), ou bien comme étant une contrainte que la solution proposée par l'algorithme de déploiement doit respecter.

La couverture des événements est exprimée par l'espace total que peut couvrir (détecter tous les phénomènes se produisant dans cet espace) un réseau de capteurs sans fil déployé. Un espace est dit couvert par un nœud quelconque, si et seulement si, cet espace se trouve dans le champ de détection de ce nœud. Plusieurs modèles de couverture ont été proposés

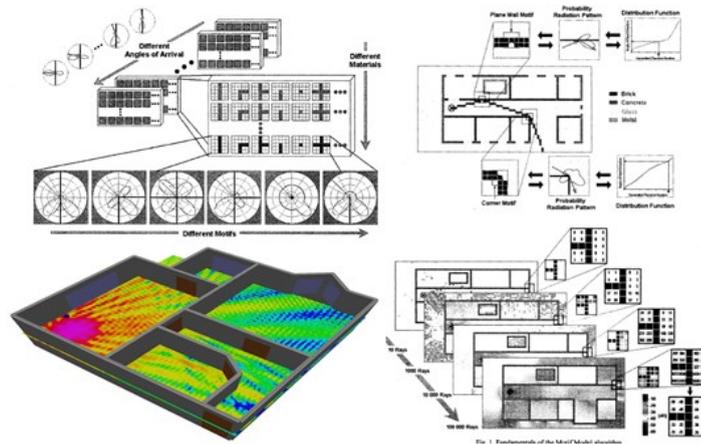


FIGURE 2.5: Quelques modèles de propagation à lancer de rayons

Le tableau Tab 2.1, présente une étude comparative entre les différents modèles cités auparavant. En vue d'avoir un modèle complet qui prend en compte tous les phénomènes liés à la propagation d'ondes radio, nous avons développé le modèle MWF-amélioré qui sera détaillé dans les prochaines sections.

Modèles	FRIIS	LOS	MWF	Lancer de Rayons	MWF-Amélioré
Avantages	Rapidité, adaptation aux espace libre	Rapidité, prise en compte de réflexion d'onde	Simplicité d'implémentation, prise en compte des obstacles	Prise en compte des phénomènes de prop. d'ondes (diffraction, réflexion, ...)	Rapidité, simplicité d'implémentation
Inconvénients	Aucune prise en compte d'obstacles	Aucune prise en compte d'obstacles	Aucune prise en compte de phénomènes (diffraction, ...)	Consommation de temps de calcul	Nécessité d'une campagne de mesures

TABLE 2.1: Tableau comparatif des modèles de propagation

dans la littérature. Selon [9] ces modèles peuvent être classés en trois sous-groupes : (1) Modèle de Détection Binaire (MDB), (2) Modèle de Détection Asymptotique (MDA), (3) Modèle de Détection Probabiliste (MDP).

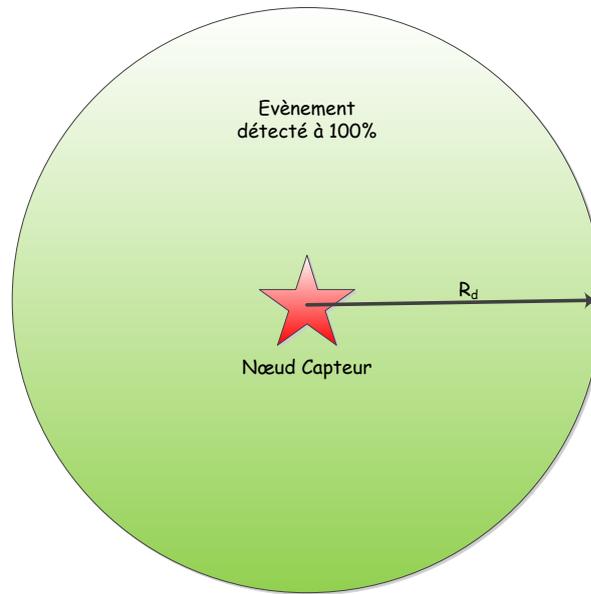


FIGURE 2.6: Modèle de Détection Binaire [8]

2.2.3.1 Modèle de Détection Binaire (MDB)

Le modèle binaire de détection est l'un des plus utilisés dans les travaux de recherche liés au déploiement d'un réseau de capteurs sans-fil. Appelé aussi modèle 0-1 [81], ce modèle considère qu'un évènement est détecté avec une probabilité égale à "1" (100% de détection) s'il se produit à une distance inférieure ou égale au rayon de détection R_d d'un nœuds capteur, et une probabilité de "0" (aucune détection) sinon (distance $> R_d$). La Figure 2.6 représente un exemple de ce modèle de détection d'évènements par un nœud capteur.

$$P_s^p = \begin{cases} 1 & \text{si } d(s,p) < R_d \\ 0 & \text{sinon } d(s,p) \geq R_d \end{cases} \quad (2.7)$$

Où P_s^p est la probabilité de détection d'un évènement qui se produit dans la position p par un capteur s , $d(s,p)$ est la distance euclidienne entre la position p et le capteur s , R_d est le rayon maximum de détection.

Ce modèle a été largement utilisé dans la littérature ([8, 81–83]) pour sa rapidité et son faible coût de calcul. L'inconvénient principal de ce modèle est qu'il n'intègre aucune notion d'obstacle. Le seul paramètre pouvant influencer la détection d'évènements est la distance séparant le lieu où l'évènement se produit et le capteur. Ce type de modèle n'est pas adapté à des applications intra-bâtiment [82], et nécessite une amélioration.

2.2.3.2 Modèle de Détection Probabiliste (MDP)

Modèle de Détection Asymptotique (MDA)

En général, la probabilité de détection d'un évènement diminue avec l'accroissement de la distance [80]. En effet, un phénomène est détecté avec une probabilité plus grande s'il se produit près du nœud capteur. Cette probabilité est ensuite réduite en fonction de la distance. Dans un modèle asymptotique, la probabilité de détection est calculée selon deux fonctions : (1) exponentielle (MDA-E), (2) polynomiale (MDA-P).

MDA-E Dans ce modèle la qualité de détection d'un évènement se dégrade exponentiellement par rapport à la distance. La probabilité de détection est donnée par l'équation suivante :

$$P_s^p = \exp^{-\alpha d(s,p)} \quad (2.8)$$

Où le paramètre α représente la qualité de détection du capteur s et qui dépend des paramètres du capteur, $d(s,p)$ représente la distance euclidienne entre le capteur s et la position p [84].

MDA-P Dans ce modèle la qualité de détection d'un évènement se dégrade d'une manière polynomiale d'ordre $(-k)$, la probabilité est donnée par l'équation :

$$P_s^p = \lambda d(s,p)^{-k} \quad (2.9)$$

Où λ est un paramètre lié aux configurations matérielles du capteur, généralement $\lambda = 1$; k est un nombre entier compris dans l'intervalle $[1, 4]$ [85, 86], et représente la qualité de détection du capteur. Une généralisation du modèle asymptotique de détection est apparue en vue de remplacer le modèle binaire. On considère dans cette méthode qu'un évènement est détecté avec une certaine probabilité qui diminue en fonction de la distance euclidienne séparant la position de l'évènement et la position du capteur. En utilisant ce modèle, on considère deux zones (cercles) de détection : (1) zone de confiance, (2) cercle maximale (limite maximale) (Figure 2.7).

Cas-1 si l'évènement se produit dans la zone de confiance (évènement (1) dans la Figure 2.7), alors il est détecté avec une probabilité égale à 1 (100% de détection).

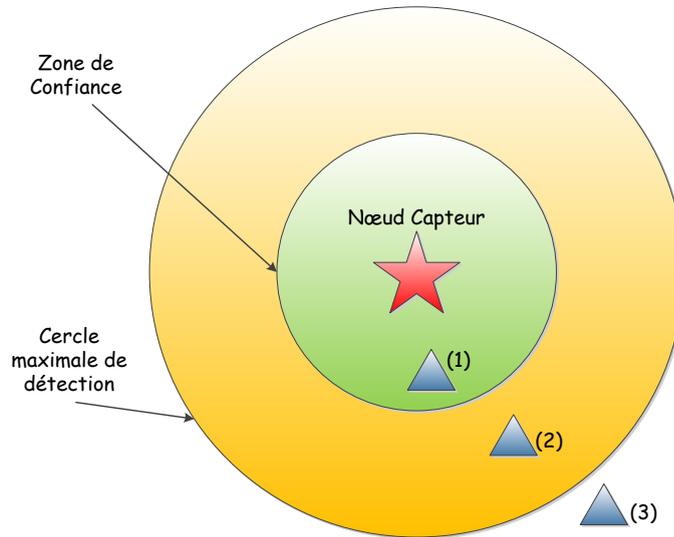


FIGURE 2.7: Modèle de Détection Probabiliste [9]

Cas-2 si l'évènement se produit entre la zone de confiance et la limite maximale (évènement (2) dans Figure 2.7) il est détecté avec une probabilité P_s^p (qui peut être une fonction exponentielle ou polynomiale) qui diminue en fonction de la distance.

Cas-3 si l'évènement se produit à l'extérieur du cercle maximal (évènement (3) de la Figure 2.7), dans ce cas l'évènement n'est pas détecté. L'équation qui définit le modèle est comme suit

$$P_s^p = \begin{cases} 1 & \text{si } d(s, p) \leq R_{min} \\ \frac{\alpha}{d(s, p)^\beta} & \text{si } d(s, p) \leq R_{max} \\ 0 & \text{si } d(s, p) \geq R_{max} \end{cases} \quad (2.10)$$

$$P_s^p = \begin{cases} 1 & \text{si } d(s, p) \leq R_{min} \\ \exp^{-\alpha d(s, p)} & \text{si } d(s, p) \leq R_{max} \\ 0 & \text{si } d(s, p) \geq R_{max} \end{cases} \quad (2.11)$$

Où R_{min}, R_{max} représentent respectivement les rayons de la zone de confiance et du cercle maximale, α est un paramètre lié aux composantes matérielles du capteur (représente le facteur de distorsion de l'énergie lié aux obstacles), β est un paramètre lié aux caractéristiques de l'évènement (toujours inférieur à 5 et dépend de l'environnement) [80, 84, 87, 88].

2.2.4 Coût de déploiement

L'un des principaux objectifs de conception d'un RCSF, d'un point de vue économique est la réduction du coût de déploiement tout en assurant d'excellentes performances [89]. Les nœuds formant le RCSF peuvent être catégorisés en deux types : (1) nœuds capteurs (collectant des données sur les phénomènes physiques), (2) nœuds routeurs (relayant ces informations jusqu'au serveur). Dans des applications ne nécessitant pas l'intervention de personnes pour l'installation et la mise en route des nœuds du RCSF, le coût de déploiement est simplement calculé par le produit du nombre de nœuds et du prix d'achat de ces derniers. Plusieurs travaux ont été conduits dans ce sens [9, 82, 90–92]. Vu que l'installation des nœuds (capteurs/routeurs) demande des efforts humains, le coût de déploiement est donc calculé selon deux facteurs : (1) coût du matériel (capteurs/routeurs), et (2) coût d'installation (Figure 2.8). Le coût d'installation varie selon l'endroit où le capteur est installé. Il peut aussi varier selon les hauteurs des points de fixations. [88]

Soit un réseau de capteurs sans-fil qui se compose de : $R = \{r_1, r_2, r_3, \dots, r_n\}$ nœuds routeurs et de $S = \{s_1, s_2, s_3, \dots, s_m\}$ nœuds capteurs, le coût de déploiement est calculé comme suit :

$$Cost_{Tot} = \sum_{i=1}^n (c_i * r_i) + \sum_{j=1}^m (c_j * s_j) \quad (2.12)$$

Où :

- r_i, s_j : sont des variables binaires qui représentent respectivement la présence (installation) ou pas d'un nœud routeur/capteur (r_i, s_j : est mise à 0 si le nœud n'est pas déployé, 1 si le nœud est installé).
- c_i, c_j : représente le coût (achat + installation) d'un nœud routeur (i), capteur (j).

2.2.5 Durée de vie du réseau

Les capteurs sans-fil sont généralement alimentés par des sources limitées en termes d'énergies (batteries ou piles). Une fois le réseau déployé, les capteurs puisent dans ces ressources afin d'assurer différentes tâches pour lesquelles ils ont été installés (transmission/réception, traitement de données, captage, ...). Une fois que cette source arrive à sa limite, le nœud capteur s'arrête de fonctionner, causant ainsi, la perte de données sur le(s) phénomène(s) qu'il était entrain de surveiller. Aussi, le dysfonctionnement de quelques nœuds peut causer des changements topologiques significatives et pourrait exiger le re-routage des paquets et la réorganisation du réseau [59].

En général, la durée de vie du réseau est représentée par la période de temps entre le déploiement du réseau et l'instant où ce dernier est considéré comme non fonctionnel. Le moment où le réseau est considéré comme non fonctionnel est, cependant,

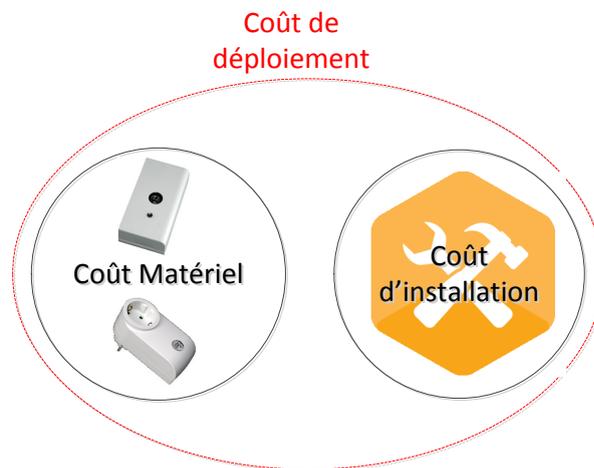


FIGURE 2.8: Modèle du coût de déploiement

spécifique à l'application. Ce moment peut être par exemple l'instant où le premier capteur consomme toute son énergie (ou bien une perte d'un ensemble de capteurs). Un partitionnement du réseau apparaît là où une perte de couverture d'une partie de la région à surveiller se produit[93].

D'après [59], un nœud capteur consomme de l'énergie quand il exécute l'une des tâches suivantes :

- *Capture (récolte d'informations) des phénomènes physiques* : la consommation d'énergie du module de captage varie selon la nature de l'application. Une surveillance aléatoire des événements consomme plus d'énergie qu'une surveillance constante. La complexité de la détection d'évènement joue aussi un rôle crucial dans la détermination de la dépense d'énergie du réseau.
- *Communication (transmission ou réception de données)* : Sont inclus dans la communication, la transmission et la réception des données. Toutes les parties du circuit de transmission (synthétiseur de fréquences, oscillateurs, amplificateurs, ...) consomment de l'énergie. Cette partie est la seule qui dépend de l'emplacement des nœuds[94]. Ceci sera exposé dans les sections suivantes.
- *traitement de données (récoltées, reçues, transmises ...)* : La dépense d'énergie du module de traitement de données est minime comparée à celle de la communication [95]. Selon [59] Le coût énergétique de la transmission de $1KBits$ à une distance de $1m$ est approximativement celle consommée pour exécuter 3 à 100 millions d'instructions par seconde sur un processeur (MIPS)/W.

Plusieurs travaux de recherches visant à prolonger la durée de vie du RCSF ont été conduits. Ces travaux portent essentiellement sur : de nouveaux protocoles efficaces en énergie d'accès au médium (MAC), des protocoles de routages efficaces en énergie, ...

Une modélisation mathématique de la durée de vie s'impose afin de pouvoir procéder à un calcul par simulation. Nous exposons dans les parties suivantes quelques travaux de recherches menés sur la modélisation de la durée de vie du RCSF.

Dans [96], les auteurs proposent un outil capable d'estimer la consommation d'énergie d'un nœud capteur. Les principales parties qui consomment de l'énergie ont été considérées dans ces travaux : le micro-processeur, le module de captage et le module de communication (transmission/réception).

Les résultats de ce travail ont montré que la composante de communication (RF) des capteurs est la plus gourmande en terme d'énergie consommée. Aussi, cette partie dépend fortement des deux paramètres : distance séparant les nœuds et débit de transfert des données. Le reste des composantes (collecte de données et leur traitement) ne représentent qu'une infime proportion dans l'énergie totale consommée par le capteur (ex. : une augmentation de 10% des données à traiter rajoute seulement 2% d'énergie consommée). Ces deux composantes dépendent du matériel utilisé lors de la fabrication des capteurs. Pour cela, nous nous intéressons dans notre travail à l'énergie consommée par la partie communication (RF) des nœuds du réseau.

Deux types de communications ont été comparés dans l'article [96] : la communication directe et multi-sauts. Dans le premier type de communication (équation 2.13) les auteurs ont proposé deux cas d'études afin d'étudier l'impact du phénomène de sur-écoute sur la durée de vie du RCSF (équation 2.14, équation 2.15). Les résultats montrent que l'énergie consommée en réception est plus grande dans le deuxième cas d'étude (prise en compte du phénomène de sur-écoute). Ceci est dû à la réception des messages envoyés par tous les nœuds voisins sans pour autant les retransmettre. Dans la deuxième expérimentation les auteurs ont comparé deux modèles de communication : linéaire (équation 2.16), et non-linéaire (équation 2.17).

2.2.5.1 Communication directe

Dans ce modèle, chaque nœud capteur communique directement avec la station de base (Sink) sans qu'aucun autre n'intervient dans l'acheminement de l'information. Dans ce schéma la communication peut se produire avec un phénomène de sur-écoute ou sans. L'énergie nécessaire pour transmettre un paquet de K bits entre deux nœuds A et B séparés par une distance d est comme suit :

Sans sur-écoute

$$E_{AB} = E_{transmit} * K * d^{\alpha} \quad (2.13)$$

Avec sur-écoute

– Sans relais

$$E_{AB} = E_{transmit} * K * d^\alpha + N * E_{receive} * K \quad (2.14)$$

– Avec relais

$$E_{AB} = E_{transmit} * K * d^\alpha + N * E_{receive} * K + \sum_{i=1}^N (E_{D(i)B}) \quad (2.15)$$

Où $D(i)$ représente chaque distance séparant le noeud i du noeud B.

2.2.5.2 Communication multi-sauts

Dans ce modèle la communication s'effectue grâce à la participation de tous les noeuds du réseau. Considérés comme des relais, ces derniers acheminent le trafic généré par les différents noeuds jusqu'à atteindre la station de base (Sink).

Modèle linéaire : Soit un réseau composé de n noeuds où chaque couple de noeuds est séparé par une distance r . Si un noeud localisé à une distance $n * r$ de la station de base veut lui transmettre un paquet de K bits, ce paquet sera relayé $(n-1)$ fois afin d'atteindre le noeud le plus proche de la station de base. Soit un noeud A situé à une distance r de la station de base émetteur d'un paquet de K bits à B, l'équation définissant l'énergie consommée par ce noeud est comme suit :

$$E_{AB} = (n - 1)(E_{transmit} * r^\alpha * K + E_{receive} * K) + E_{transmit} * K \quad (2.16)$$

Modèle non-linéaire Dans une topologie réseau non-linéaire dense, chaque noeud est prévenu par le protocole de routage si un acheminement est nécessaire ou pas, un phénomène de sur-écoute se produit (paquets reçus mais non acheminés). Ce modèle est défini par l'équation suivante :

$$E_{AB} = E_{transmit} * K * n * d^\alpha + E_{receive} * K * L \quad (2.17)$$

Où :

- α : représente le facteur d'atténuation lié à l'environnement de déploiement ;
- $E_{transmit/receive}$: l'énergie consommée par le circuit de transmission et de réception ;
- n : nombre de noeuds participant à l'acheminement de l'information (paquets) jusqu'à la station de base ;
- L : nombre de noeuds recevant des paquets de noeuds voisins sans les acheminer (sur-écoute).

D'après [97], la durée de vie est définie comme étant le nombre d'unités de temps pendant lesquelles le RCSF peut collecter des informations sur son environnement et les transmettre. La consommation d'énergie au temps t est définie par l'équation suivante :

$$E_{i(t)} = K * (r_i(t) + 1) * \beta * d^\alpha(i, e) * amp \quad (2.18)$$

Où $E_{i(t)}$ représente la consommation d'énergie du noeud i au temps t , K est le nombre de bit transmis (taille du paquet), $r_i(t)$ représente le nombre de paquets à retransmettre au temps t , β est un paramètre lié à la qualité de transmission, α est le facteur d'atténuation lié à l'environnement (atténuation de l'onde) et amp représente l'énergie consommée par l'amplificateur pour transmettre un seul bit.

Dans [98], les auteurs ont développé un module de calcul de la consommation d'énergie d'un RCSF, qui a été intégré dans le simulateur Prowler. Aussi, des modifications de la couche MAC dans Prowler ont été apportées en vue d'atteindre les mêmes performances que celles du simulateur TinyOs. Le simulateur développé (Prowler) ne calcule que les temps passés dans chaque état avec une prise en compte des collisions et erreurs lors de la transmission des données. Par conséquent, la consommation d'énergie dans ce modèle est calculée par le biais de l'équation suivante :

$$E_{Tot} = \sum_{etatj} (P_{etatj}) * t_{etatj} + \sum (E_{transition}) \quad (2.19)$$

où :

- P_{etatj} : représente la puissance dans l'état j ;
- $E_{transition}$: représente l'énergie consommée par la transition d'un état à un autre.

Dans [99], les auteurs présentent une analyse de la consommation d'énergie d'un RCSF opérant sous le protocole IEEE 802.15.4 (Zigbee). La durée de vie est définie comme étant le nombre de fois où le noeud peut utiliser la même stratégie de communication (exemple : si $E_0 = 400mAh$ et que pendant une heure le capteur consomme $100mAh$, alors la durée de vie sera égale à 4 heures $LT = (\frac{400}{100}) * 1heure$). La consommation d'énergie est calculée en se basant sur le processus de communication typique du protocole Zigbee comme le montre la Figure 2.9, et elle est donnée par les équations suivantes : La durée totale est donnée par :

$$T_{TX_MAC_ACK} = D_{SDP} + D_{RX,L} + D_{R2T} + D_{TX,D} + D_{T2R} + D_{RX,M} + D_{PS} \quad (2.20)$$

L'énergie consommée pendant cette durée est donnée par :

$$E_{TX_MAC_ACK} = \frac{P_{SDP}D_{SDP} + P_{RX}(D_{RX,L} + D_{RX,M}) + P_{R2T}D_{R2T}}{T_{TX_MAC_ACK}} + \frac{P_{TX}D_{TX,D} + C_{T2R}D_{T2R} + C_{PS}D_{PS}}{T_{TX_MAC_ACK}} \quad (2.21)$$

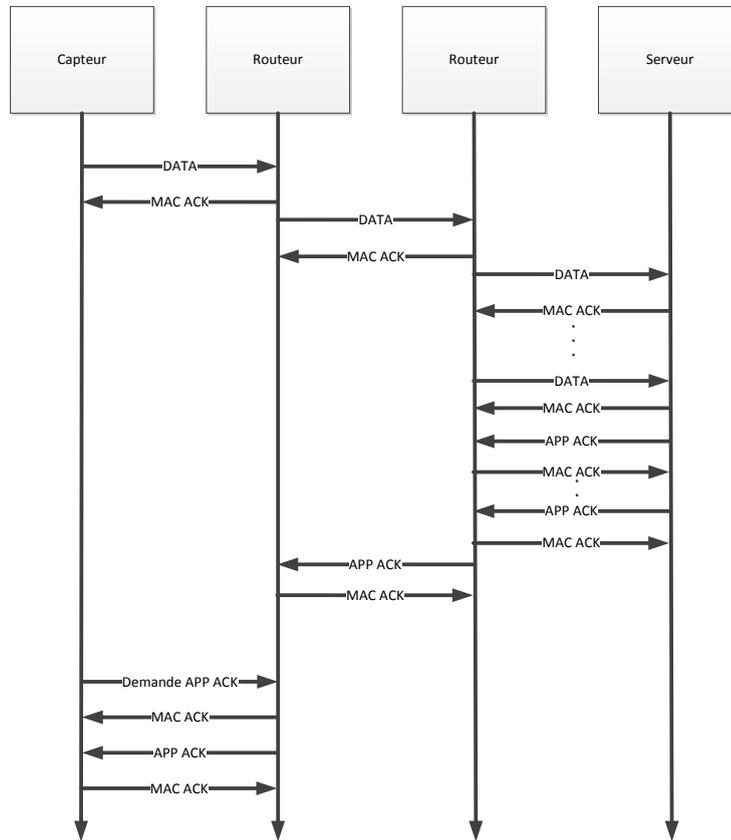


FIGURE 2.9: Processus de reporting de données dans des RCSF opérant sous Zigbee

La durée de vie est ensuite calculée en suivant l'équation suivante :

$$LT_{MAC_ACK} = \frac{BC - C_{WA}D_{WA}}{(S - T_{TX_MAC_ACK})C_S + T_{TX_MAC_ACK}C_{TX_MAC_ACK}} \quad (2.22)$$

Les différentes variables utilisées sont définies dans la Figure 2.10. Un autre travail a été conduit par Tabelsi Z. et al dans [100], qui présente une analyse des modèles de communication d'énergie d'un RCSF opérant sous le protocole IEEE 802.15.4 (Zigbee). Dans ce protocole, il existe deux types de dispositifs : (1) FFD (Full-Function Devices), (2) RFD (Reduced-Function Devices). Dans le premier type on peut retrouver : (1) PAN (Personal Area Coordinator), (2) Coordinateur, (3) End Devices (capteurs). Le deuxième type de dispositifs ne communique qu'avec le PAN, il est considéré comme étant le nœud final du réseau et agit comme un collecteur de donnée généré par la totalité du réseau. Selon les auteurs, la durée de vie pourrait être définie comme étant la durée entre la première opération effectuée par le réseau et le temps où :

- Le premier nœud dans le réseau tombe en panne d'énergie ;
- Le réseau est divisé en deux ou plusieurs parties ;
- Perte de la moitié du réseau ;

Operations		Current Consumption	Time
Waking up		C_{WA}	D_{WA}
Setting up the radio, dealing with the packet through ZigBee stack, and preparing for transmission		C_{SDP}	D_{SDP}
Receiving	Data	C_{RX}	$D_{RX,D}$
	MAC acknowledgement		$D_{RX,M}$
	APP acknowledgement		$D_{RX,A}$
	Listen		$D_{RX,L}$
Switching from receiving mode to transmission mode		C_{R2T}	D_{R2T}
Transmission	Data	C_{TX}	$D_{TX,D}$
	MAC acknowledgement		$D_{TX,M}$
	Data request		$D_{TX,DQ}$
Switching from transmission mode to receiving mode		C_{T2R}	D_{T2R}
Waiting to request the application acknowledgement when a sensor node enters an IDLE state		C_{WI}	D_{WI}
Preparing to sleep		C_{PS}	D_{PS}
Sleeping		C_S	-----

FIGURE 2.10: Variables utilisées [10]

– Perte de l'information sur un ou plusieurs phénomènes.

L'énergie consommée par un nœud est exprimée par la somme des énergies consommées en : transmission, réception, sur-écoute, idle (écoute), overhead (transmission/réception des paquets de contrôle).

Les auteurs ont aussi introduit dans leur modèle la notion de pertes des paquets dues aux collisions et erreurs de transmission, ce qui génère une retransmission du même paquet, et donc un sur-coût d'énergie. Le nombre de paquets générés par un nœud est calculé par rapport aux deux probabilités : collision et erreur. Dans [101], les auteurs proposent une modélisation réaliste des différentes puissances consommées par un nœud capteur. Les auteurs n'ont pris en compte que les pertes du signal liées à une propagation en espace-libre où l'onde transmise se dégrade avec l'accroissement de la distance avec un facteur d'atténuation égal à 2 (typique en environnement free-space). Ce travail a été la base des recherches faites dans [102] sur la consommation d'énergie d'un nœud capteur. Les auteurs ne considèrent que les consommations d'énergie générées par le système de communication (transmission/réception). Cette consommation peut être divisée en deux parties : (1) consommation en mode réception, (2) consommation en mode transmission. Les auteurs mettent en avant le lien entre les puissances consommées en

transmission/réception et les énergies consommées par le module de communication. Les auteurs se sont basés sur le mode opératoire d'une transmission entre deux nœuds capteurs. Un tableau est présenté dans leur article regroupant les différentes étapes nécessaires au système de communication avant de pouvoir transmettre ou recevoir des données. Chaque étape est étudiée séparément et l'énergie totale consommée par un nœud pendant une période T est donnée par la somme des énergies consommées par les différents états du système de communication.

Dans [103], les auteurs ont présenté un modèle typique de la consommation énergétique d'un module de transmission d'un nœud capteur. La consommation des différentes parties a été considérée comme étant constante et ne dépend que des circuits électriques des sous-systèmes. Les auteurs ont concentré leurs travaux sur la consommation d'énergie d'un nœud capteur (dans le mode transmission) utilisant la modulation *FSK*. L'énergie consommée en transmission a été représentée par la somme des énergies consommées par les différentes composantes du bloc analogique d'un transmetteur-récepteur *FSK* selon l'équation suivante :

$$E_{ana_L} = E_t + E_{amp} + E_{c_{ana}} + E_{tr} \quad (2.23)$$

En rajoutant l'énergie consommée par le détecteur d'évènement de la modulation *FSK* à l'équation suivante, les auteurs ont obtenu la valeur totale consommée pour émettre un bit de donnée :

$$E_{Tot} = \frac{P_t T_{on} + a P_t T_{on} + P_c T_{on} + 2 P_{syn} T_{syn} + P_{FSK_Detect} T_{on}}{L} \quad (2.24)$$

$$P_c = P_{c_{ana}} + P_{c_{dig}} \quad (2.25)$$

Finalement, les auteurs ont présenté un compromis entre la bande passante et le temps de transmission. Aussi, l'énergie consommée pour émettre un bit de donnée a été modélisée analytiquement.

Les auteurs dans [104], présentent un modèle de consommation d'énergie d'un RSCF dans des réseaux adoptant une topologie Cluster. On prend en considération deux types de nœuds : (1) des nœuds normaux, (2) des nœuds surcouches qui sont plus efficaces en énergie. Tous les nœuds participent à la collecte des données sur les phénomènes physiques et envoient leur rapport par la suite à la tête du cluster qui fait office de relais entre les différents capteurs et le serveur (Sink). Dans leur modèle de consommation d'énergie, les auteurs ont pris en considération les énergies consommées par :

- La transmission/réception d'un bit de donnée ;
- Le captage d'un bit de donnée ;
- Le décodage d'un paquet.

L'énergie totale consommée est exprimée par la somme de toutes les énergies. Notons que seule l'énergie consommée par le module transmetteur varie au dépend de la distance qui

sépare le transmetteur, le récepteur et les paramètres du modèle de propagation adopté (l'espace-libre dans l'article).

A partir de toutes ces définitions nous pouvons dire que la durée de vie est représentée par le nombre totale d'unités de temps passées entre la première opération qu'effectue le RCSF et le temps où un nœud (ou une partie du réseau) perd toute son énergie initiale. Aussi, nous avons relevé que le module de communication est celui qui consomme le plus d'énergie dans un nœud capteur. La consommation d'énergie du module de communication (transmission/réception) varie selon la distance séparant deux nœuds communicants.

Nous pouvons dire que la consommation d'énergie du module de communication dépend principalement du positionnement des nœuds et du nombre de paquets à transmettre. Pour cela, nous nous intéresserons à l'énergie consommée par le module de communication (RF). Les consommations d'énergie des modules restants seront considérées comme étant constantes.

2.3 Modèles retenus et expérimentations

Dans cette partie du travail, Nous présenterons les modèles retenus et les premières expérimentations menées dans notre laboratoire. Ces expérimentations porteront essentiellement sur les aspects de communication et de durée de vie du RCSF. Toutes les expériences menées dans ce chapitre se basent sur le même cas d'étude avec plusieurs scénarios. Ce choix est retenu afin de pouvoir établir une étude comparative entre les différents scénarios proposés.

Les deux expérimentations valident le choix du modèle de propagation et montrent l'impact du positionnement des nœuds du réseau sur sa durée de vie.

2.3.1 Préliminaires et hypothèses

Notre méthode de déploiement utilise un modèle de propagation adapté aux environnements intérieurs (*MWF* Multi-Wall Floor propagation model) qui a été décrit dans la section 2.2. Dans ce modèle le *RSSI* (Received Signal Strength Indicator) est calculé en soustrayant les atténuations provoquées par les différents obstacles (murs, fenêtres, ...) de la puissance de l'onde propagée par la source (nœud capteur ou routeur).

Pour ce qui est de la couverture, un point couvert par un nœud s_i est défini comme suit :

Définition

Point couvert : un point quelconque p appartenant à l'espace de déploiement A est dit

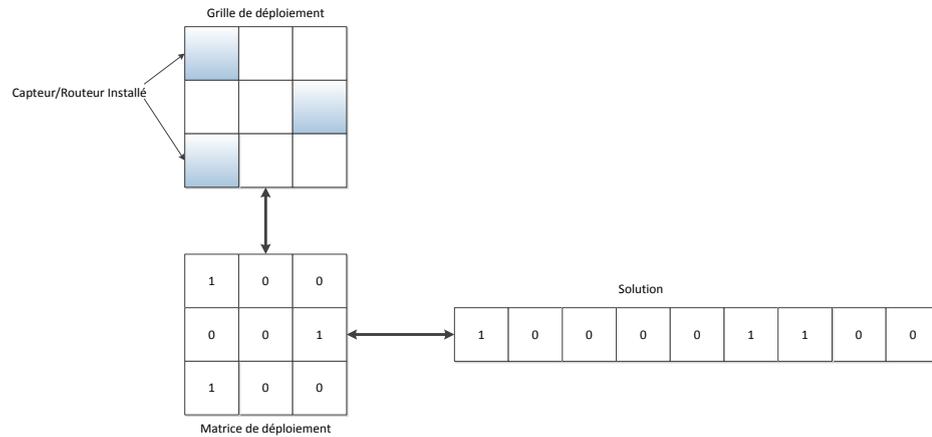


FIGURE 2.11: Processus de retranscription Grille-Solution

couvert par un nœud s_i si et seulement si, p appartient au périmètre de captage du nœud s_i et aucun obstacle n'existe entre p et s_i . Cette définition peut être représentée mathématiquement comme suit :

$$\phi_{s_i} = \{p | p \in \text{sphère}(s_i, R_i) \wedge !O(\vec{ps}_i)\} \quad (2.26)$$

Où ϕ_{s_i} représente l'ensemble des points couverts par le nœud s_i . $\text{sphère}(s_i, R_i)$ représente une sphère d'un rayon R_i et centrée au nœud s_i , elle indique l'espace de captage du nœud s_i . $O(\vec{ps}_i)$ est une variable binaire qui indique la présence (1), ou pas (0), d'un obstacle entre le nœud s_i et le point p .

2.3.2 Codage de la solution

Dans l'objectif d'inclure à la fois la position et le coût dans une seule solution, nous avons choisi de représenter cette dernière par un vecteur binaire. Premièrement, la grille de déploiement (section 2.2) est traduite en une matrice binaire D , où chaque variable représente la présence (ou pas) d'un nœud (capteur/routeur) dans la cellule (i,j) . Par la suite, la matrice D est retranscrite en un vecteur (utilisée comme solution) comme le montre la figure 3.6. Un algorithme de retranscription, qui retraduit la solution en matrice de déploiement et *vice-versa* est inclu dans l'algorithme d'évaluation. Avec ce codage de solution, nous pouvons avoir les positions des nœuds déployés (capteurs/-routeurs) représentés par les indexes (i,j) de la matrice D . Le coût de déploiement est simplement calculé en sommant les éléments qui composent la matrice.

2.3.3 Coût du RCSF

L'un des objectifs de conception d'un RCSF d'un point de vue économique est de réduire le coût tout en satisfaisant les exigences d'une application. Plusieurs compagnies et organismes de recherche ont vu le jour afin de concevoir et de fabriquer des nœuds capteurs, qui fournissent plusieurs options de budget. Par ailleurs, l'installation d'un nœud capteur requiert des efforts humains supplémentaires. Pour l'instant, le coût de placement d'un capteur dans des plafonds ou murs est différent de celui du placement des nœuds dans des espaces humains actifs. Comme résultat, nous pouvons diviser le coût d'un RCSF en : coût matériel et coût d'installation.

Nous considérons dans notre travail un RCSF composé d'un ensemble $S = s_1, s_2, \dots, s_N$ de nœuds capteurs et un ensemble $SK = Sk_1, Sk_2, \dots, Sk_N$ de nœuds routeurs (Sink). Nous émettons l'hypothèse que les nœuds capteurs ont le même prix d'achat, ainsi que les nœuds routeurs. Le coût de déploiement du réseau est calculé selon la fonction suivante :

$$Cost = C_s * ||S|| + C_{Sk} * ||SK|| \quad (2.27)$$

C_s, C_{Sk} : représentent, respectivement le coût d'achat et d'installation d'un nœud capteur (routeur). Ces coûts sont considérés comme étant constants.

$||S||, ||SK||$: sont les cardinalités des ensembles S et SK.

En se basant sur la définition de l'espace de déploiement (voir section précédente), nous pouvons définir la matrice de déploiement D comme suit :

$$D(i, j) = \begin{cases} 1 & \text{si un capteur - routeur est installé dans la position } (i, j) \\ 0 & \text{sinon} \end{cases} \quad (2.28)$$

Ainsi, la fonction objective exprimant le coût de déploiement est définie comme suit :

$$Cost_{Tot}(Type) = \sum_i \sum_j D(i, j) * C_{achat} * C_{installation} \quad (2.29)$$

Où $Type$ peut être : capteur ou bien routeur. $C_{achat/installation}$ représente le coût d'achat (et installation) du nœud déployé dans la cellule (i,j).

2.3.4 Couverture

L'un des facteurs les plus importants à tenir en compte lors du déploiement d'un RCSF est sa capacité à collecter les informations sur les événements physiques environnants. Cette propriété est exprimée par l'espace total couvert par les nœuds capteurs déployés. Le concept de couverture peut être catégorisé en se basant sur le niveau de densité des

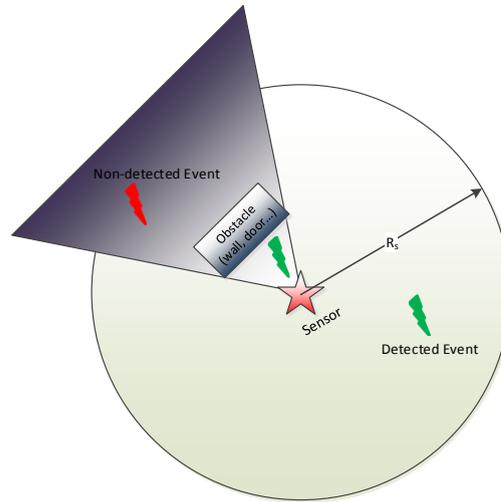


FIGURE 2.12: Modèle de couverture d'évènement

nœuds déployés. Si seulement une partie du bâtiment est couverte par l'ensemble des nœuds capteurs déployés, la couverture est dite *incomplète*, *clairsemée*. Si le bâtiment est complètement couvert par l'ensemble des nœuds la couverture est dite *dense*. Par ailleurs, si une cellule (ou plusieurs) est couverte par plusieurs nœuds, la couverture est dite *redondante*.

En règle générale, la couverture peut être définie comme étant le pourcentage de l'espace de déploiement où un évènement peut être surveillé par (au minimum) un nœud capteur. En toute évidence, dans un déploiement parfait, la couverture doit être égale à 100%. Sachant qu'un nœud capteur peut être endommagé ou bien perdre toute son énergie (et ainsi s'éteindre), nous proposons d'utiliser la notion de *k-coverage* ($k \geq 1$), où chaque cellule du bâtiment peut être surveillée par au moins k nœuds capteurs.

Définition de la k-couverture *A n'importe quel moment t , n'importe quel point $p \in A$ doit être couvert par k différents capteurs de l'ensemble S [105]*

Se basant sur cette définition nous pouvons déduire la fonction C_A qui représente le pourcentage de l'espace couvert par l'équation suivante :

$$C_A(S) = \frac{|\sum_{i=1}^N \{p | p \in A \wedge p \in \phi_{s_i}\}|}{|\{p | p \in A\}| * k} \quad (2.30)$$

Afin de simplifier les calculs nous proposons l'utilisation du modèle binaire (vue dans le chapitre précédent). Nous intégrons une nouvelle fonction au modèle afin de prendre en compte les obstacles. Un nœud capteur ne peut pas détecter les évènements qui se produisent dans une cellule si un obstacle existe. La Figure 2.12 représente une vue globale du modèle de couverture proposé. Grâce à cette définition, nous pouvons exprimer la matrice de couverture par :

$$Couv_A(i, j) = \begin{cases} 1 & \text{Si la cellule } (i, j) \text{ est couverte par au moins } k \text{ capteurs de } S \\ 0 & \text{sinon} \end{cases} \quad (2.31)$$

Finally, the objective function representing coverage is represented by the following equation :

$$Couv_A(S) = \frac{|\sum_{i=1}^L \sum_{j=1}^W Cov_A(i, j)|}{L * W} * 100 \quad (2.32)$$

2.3.5 Connectivité

The main task of a node receiver is to detect events that occur in its environment and to transmit them in turn to a node relay or router (Sink). If the node router (Sink) is within the communication range of the node receiver, the event can be easily transmitted. On the other hand, if the node router is not within the communication range of the node that detects the event, the information is lost. Thanks to this definition, the RCSF can be represented as a graph $G=(V, E)$ connected, where V is the set of deployed nodes, $E \subseteq V^2$ is the set of links connecting them. If a link (N_u, N_v) or $(N_v, N_u) \in E$ exists, it means that the two nodes N_u, N_v can establish a communication. The set $\eta(u)$ representing the nodes adjacent to node u can be expressed as follows :

$$\eta(u) = \{N_v | (N_u, N_v) \in E \vee (N_v, N_u) \in E\} \quad (2.33)$$

We have seen in section 2.2 that a wireless link is determined as a function of the received signal power which is calculated using the MWM model (section 2.2.2.3), thus, the set of links can be defined as follows :

$$E = \{(N_u, N_v) \in V^2 | u \neq v \wedge RSSI(uv) \geq RX_{Tsh}\} \quad (2.34)$$

$RSSI(uv)$ represents the received signal power at node N_v and emitted by node N_u and RX_{Tsh} is the sensitivity of the receiving antenna.

2.3.6 Durée de vie

According to the works seen in section 2.2, the lifetime is expressed by the total time spent between the first operation performed by the RCSF and the moment when the first receiver/router loses all its initial energy (battery/accumulator). The development of a simulator for RCSF that can recover this information is indispensable. For this,

nous intégrons dans notre programme une partie simulation du RCSF qui se base sur le protocole LEACH [106].

2.4 Expérimentations

Dans le cadre du projet CREST (Community REtrofit through Sustainable Technology), nous avons développé un living-lab composé de plusieurs capteurs sans-fil permettant d'acquérir des données sur les usages du bâtiment. Les principaux capteurs sans-fil acquis dans le laboratoire sont comme suit : température et humidité (ZHum), présence (ZMove), consommation d'électricité (ZPlug) et luminosité (ZLum). Le living Lab développé servira de banc de test pour des applications de type bâtiment intelligent, il nous permettra par la suite la récolte d'informations sur les usages des appareils consommateurs d'énergie électrique. Ces informations serviront de point de départ pour le développement de système de gestion automatique du bâtiment. Dans ce qui suit, nous détaillerons les caractéristiques des capteurs acquis, ainsi que, les expérimentations menées au sein du laboratoire. L'objectif de ces expériences est de valider les modèles vus auparavant. Aussi, une étude comparative de la durée de vie des RCSF est exposée dans cette partie.

2.4.0.1 Capteurs retenus

Les différents capteurs retenus sont conçus et fabriqués par l'entreprise Française Cleode. Cette dernière propose une large gamme de produits innovants qui sont conformes à la norme Zigbee et plus précisément à la norme Zigbee Home Automation dédiée aux applications "bâtiments intelligents".

- *Température et Humidité (ZHum)* : Le ZHum est un capteur sans-fil qui mesure la température et l'humidité relative dans la pièce où il est installé. (voir la Figure 2.13 (A))
- *Présence (ZMove)* : Le ZMove est un capteur de présence qui contient un capteur infrarouge permettant de détecter les mouvements des personnes dans une pièce. Ce capteur peut détecter les mouvements dans un périmètre inférieur ou égal à 10 mètres. Il contient une application de détection de type *Occupancy Sensor* définie par la norme Zigbee Home Automation qui lui permet de se connecter à d'autres dispositifs du même type. Il dispose d'un angle horizontal de 110 degré et vertical de 93 degré (voir la Figure 2.13 (B))
- *Consommation d'électricité (ZPlug)* : Le ZPlug est un capteur sans-fil sous forme de prise électrique qui peut mesurer la consommation électrique des appareils qui sont branchés dessus. Il accepte le branchement d'appareils consommant jusqu'à 3500

Watts et fournit leurs consommations électriques en temps-réel. Ce dispositif est différent des autres car il sert aussi de routeur (Sink), ce qui lui permet d'étendre le réseau en reliant les informations reçues (des autres types de capteurs) et le serveur. Aussi, ce capteur intègre une option On/Off qui nous permet d'éteindre (ou d'allumer) les appareils branchés à distance. (voir la Figure 2.13 (C))

- *Luminosité (ZLum)* : Le ZLum est un dispositif permettant d'acquérir des données sur le niveau de luminosité dans une pièce. Ce capteur opère sous le protocole Zigbee et peut être connecté aux autres éléments du réseau (capteurs mentionnés ci-dessus). Il envoie des notifications sur le niveau de luminosité au système de gestion du bâtiment, ce qui permet par la suite d'éteindre ou d'allumer les lumières dans l'objectif d'efficacité énergétique. (voir la Figure 2.13 (D))

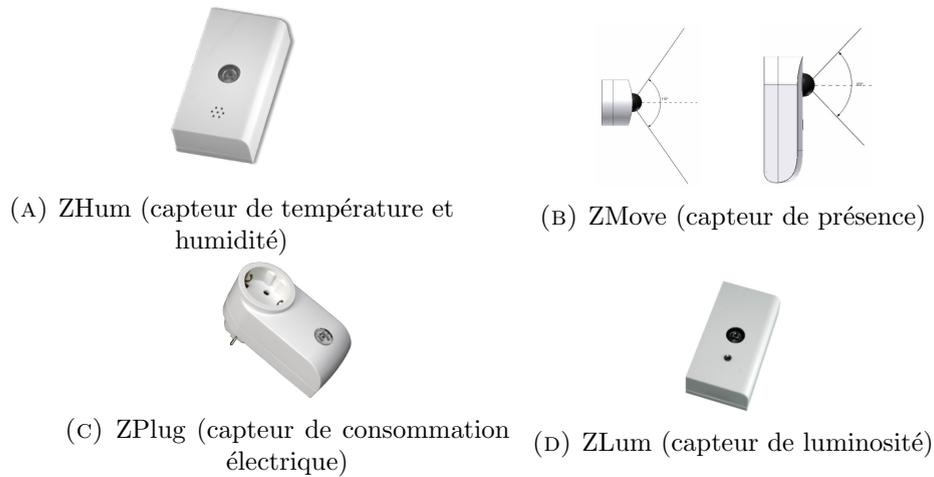


FIGURE 2.13: Capteurs implémentés dans le Living-Lab du CESI

Un résumé des principales caractéristiques des différents capteurs mentionnés ci-dessus est présenté dans le tableau suivant :

Capteur	Poids	Autonomie	Rayon	Connectivité	Mesure	Allimentation
ZHum	30 g	$\leq 1an$	—	100/30 m (espace libre/intérieur)	0-100; -40 °; 125 °	Pile (3V)
ZMove	70 g	$\leq 1an$	10m	100/30 m (espace libre/intérieur)	—	Pile (3V)
ZPlug	150 g	—	—	16A et 220V	100/30 m (espace libre/intérieur)	secteur (220V)
ZLum	30 g	$\leq 1an$	—	100/30 m (espace libre/intérieur)	1Lux-655362Lux	Pile (3V)

TABLE 2.2: Tableau récapitulatif des capteurs utilisés

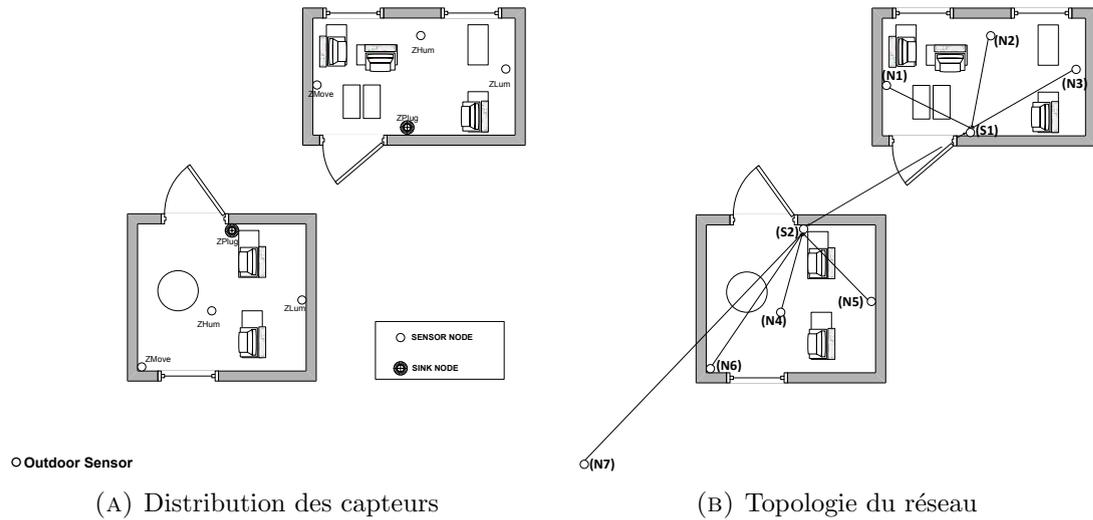


FIGURE 2.14: Architecture générale du bâtiment

2.4.1 Puissance de transmission vs. distance

2.4.1.1 Cas d'étude

Dans l'objectif d'étudier l'importance du choix du modèle de propagation radio et l'impact sur la consommation énergétique des différents capteurs, nous avons élaboré un scénario de simulation. Notre première étude s'est résumée à une partie restreinte du bâtiment CESI-Recherche, qui consiste en deux salles (bureaux) séparées par plusieurs cloisons Figure 2.14a. Chaque salle est équipée de plusieurs capteurs comme suit :

- Un capteur de température/humidité (ZHum) ;
- Un capteur de présence (ZMove) ;
- Un capteur de consommation électrique (ZPlug) qui fait office de nœud routeur (Sink).

La topologie du réseau déployé est représenté dans la Figure 2.14b. Un capteur de température/humidité est utilisé à l'extérieur pour avoir un rapport de température/humidité extérieur/intérieur Figure 2.14a.

2.4.1.2 Modèle de propagation de FRIIS

Afin d'identifier les principaux facteurs influencés par le positionnement d'un nœud capteur, nous avons mis en place une stratégie de simulation de RCSF. Nous commençons par l'étude de l'impact de la distance séparant deux nœuds sur les puissances consommées par ces derniers. Nous adoptons le modèle de propagation FRIIS (espace libre) mentionné auparavant (Section 2.2.2). Dans ce modèle, l'onde radio transmise par un nœud se propage d'une manière homogène en forme de sphère de rayon R , r (avec

$R > r$) et de centre O . Chaque noeud se trouvant dans ce périmètre (sphère) peut communiquer (envoyer/recevoir des données) avec le noeud se trouvant dans le centre O . La perte de signal est calculée selon l'équation 2.3.

Résultats En utilisant l'équation qui définit le modèle de FRIIS, nous calculons la puissance de transmission nécessaire pour une connectivité optimale à plusieurs distance. Pour ce faire, nous utilisons deux noeuds (un capteur, et un routeur) séparés d'une distance d qui varie de $10m$ à $150m$. Le tableau 2.3 et la Figure 2.15 résument les différentes valeurs obtenues par le calcul théorique.

Distance (m)	Puissance de transmission (Watt)
10	1.0528e-05
30	9.4748e-05
50	2.6319e-04
70	5.1585e-04
90	8.5273e-04
110	1.3e-03
130	1.8e-03
150	2.4e-03

TABLE 2.3: Puissance de transmission vs. Distance

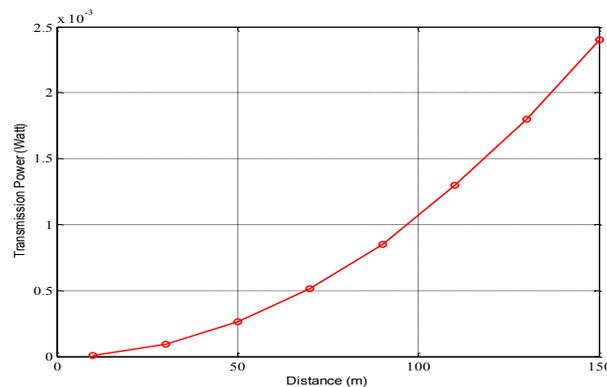


FIGURE 2.15: Puissance de transmission vs. Distance

Les résultats obtenus à ce stade du travail nous montrent que la puissance de transmission augmente en fonction de la croissance de la distance. En se basant sur ces résultats (Tab 2.3, Figure 2.15) nous développons un script TCL sous le simulateur réseau NS 2.35 (Network Simulator), afin de déterminer les puissances consommées par les deux modules Transmission/Réception. Dans le script développé, chaque capteur envoie périodiquement (chaque 5 min) une notification au noeud routeur. Le Tab 2.4 indique les paramètres utilisés dans la simulation sous NS 2.35 (taille du paquet, débit,

...). Les énergies consommées par les modules restants (à savoir celles consommées par le module de traitement des données) sont considérées comme étant constantes et ne dépendent pas de la distance séparant les noeuds. Nous avons utilisé donc le protocole *Zigbee (IEEE 802.15.44)* comme protocole de communication des noeuds. La topologie adoptée consiste en une topologie en étoile où tous les noeuds capteurs (ZHum, ZLum, ZMove) envoient périodiquement des notifications au routeur (ZPlug). Nous incluons dans le script TCL un modèle de propagation espace libre (FRIIS) et le protocole de routage AODV avec un paquet de taille 128 *bits* et un débit de 256 Kb par seconde.

Les résultats obtenus par la simulation sous NS 2.35 sont représentés par la Figure

<i>Paramètres</i>	<i>Valeurs numériques</i>
Débit	256Kb/s
Taille du paquet	128 <i>bits</i>
Sensibilité de l'antenne de réception	-90 dBm
Protocole de routage	AODV
Protocole de communication	Zigbee
Modèle de propagation	Espace-Libre
Période de notification	5 min

TABLE 2.4: Récapitulatif des paramètres utilisés dans NS 2.35

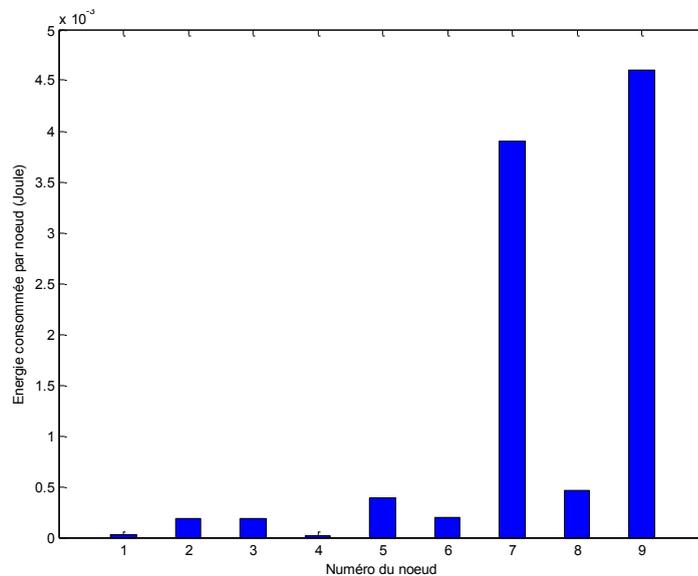


FIGURE 2.16: Énergie consommée par chaque noeud du réseau

2.16 et le tableau Tab 2.5. Nous pouvons voir que les noeuds qui consomment le plus d'énergie sont les noeuds routeur S1 (ZPlug salle 1) et le noeud N7 (capteur humidité et température extérieure). Nous pouvons déduire que :

- Pour le noeud routeur S_1 , la consommation excessive d'énergie est due au nombre des notifications qui y transitent. Ce noeud est le seul connecté à plusieurs dispositifs du réseau. Ceci nous ramène à la conclusion suivante : *Plus le nombre de noeuds connectés est grand, plus la consommation d'énergie grandit et plus la durée de vie du RCSF diminue.* Il faut donc, inclure cette contrainte lors du développement du modèle de déploiement du réseau de capteurs sans-fil.
- Pour le noeud N_7 , la consommation excessive d'énergie est totalement due à la distance le séparant du plus proche noeud routeur (S_2). Grâce à ce résultat, nous pouvons conclure que : *Plus la distance est grande plus la consommation d'énergie grandit.* Nous pouvons retenir que la distance séparant les noeuds communicants est un paramètre important qui influe sur la durée de vie du réseau.

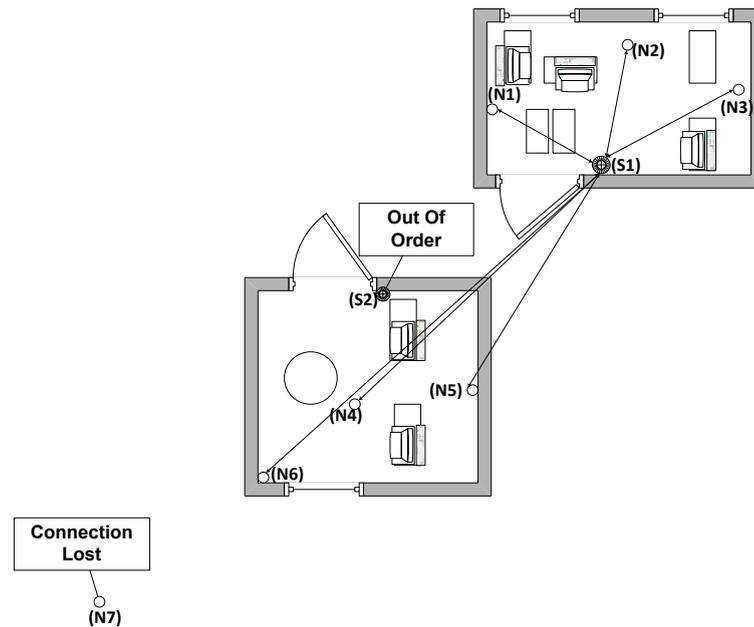


FIGURE 2.17: Topologie réseau du deuxième scénario

Noeud	E_{Tx} (Joule)	E_{Rx} (Joule)
N_1	2.2128e-05	1.8432e-12
N_2	1.8115e-04	2.1357e-12
N_3	1.8315e-04	1.0520e-12
N_4	1.9128e-05	1.6818e-12
N_5	3.9319e-04	1.7952e-12
N_6	2.0015e-04	2.2335e-12
N_7	3.9e-03	1.9513e-12
$S_1(N_8)$	4.5788e-04	6.8190e-12
$S_2(N_9)$	4.6e-03	7.8653e-12

TABLE 2.5: Énergies consommées par chaque noeud résultat NS 2.35 (scénario-1)

Dans le but d'étudier l'impact du nombre de noeuds routeurs (Sink) sur la durée de vie du réseau, nous simulons une panne se produisant au niveau d'un noeud routeur (S_2). Tous

les messages générés par les capteurs qui étaient connectés aux routeur S_2 , et qui sont dans le champs de communication du noeud routeur S_1 , vont rediriger leurs données vers ce dernier. Nous remarquons une déconnexion du noeud capteur humidité/température extérieur (N_7) qui ne peut se connecter au noeud routeur S_1 , ceci est du à la distance très importante les séparant. La Figure 2.17 montre la nouvelle topologie du réseau.

Les résultats obtenus lors de la simulation sous NS-2.35 du deuxième scénario (panne

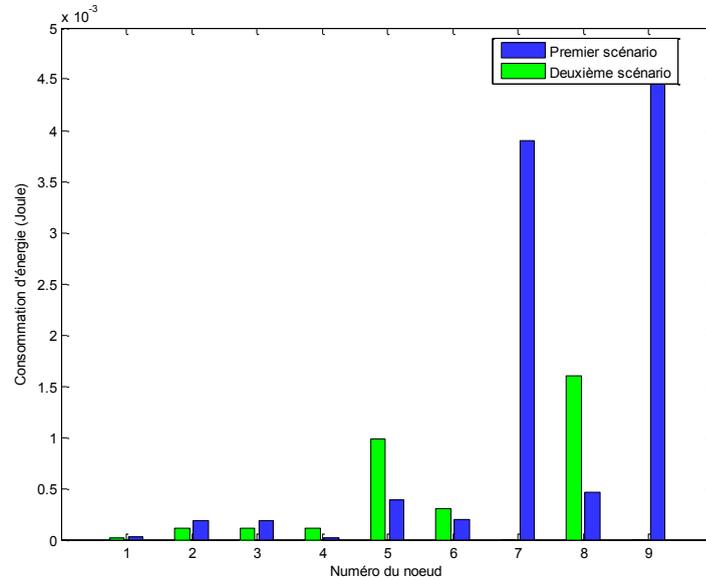


FIGURE 2.18: Énergie consommée par chaque noeud du réseau (premier et deuxième scénario)

Noeud	E_{Tx} (Joule)	E_{Rx} (Joule)
N_1	1.2128e-05	1.1520e-12
N_2	1.0915e-04	1.1520e-12
N_3	1.0915e-04	1.1520e-12
N_4	1.0915e-04	1.1520e-12
N_5	9.8234e-04	1.1520e-12
N_6	3.0319e-04	1.1520e-12
N_7	—	—
$S_1(N_8)$	1.6e-03	6.9120e-12
$S_2(N_9)$	—	—

TABLE 2.6: Énergies consommées par chaque noeuds (deuxième scénario)

au niveau du routeur S_2) sont exprimés par le Tableau Tab 2.6 et la Figure 2.18 . Nous remarquons une augmentation de la consommation d'énergie au niveau du noeud routeur (S_1), qui se voit multiplier sa consommation par trois, ceci est du au trafic de données très dense qu'il récolte. En effet, avec la perte du deuxième noeud routeur (S_2) tous les dispositifs présents dans la deuxième salle se connectent au routeur S_1 .

Nous remarquons aussi la perte des données collectées par le noeud N_7 , ceci est dû à la distance le séparant du noeud S_1 . En effet, cette perte n'est pas négligeable car toutes les informations extérieures récoltées sont perdues ; ceci peut causer le dysfonctionnement du système de gestion automatique du bâtiment d'une manière totale, car ce système ne peut plus percevoir l'environnement extérieur. Sans ces informations il est difficile de savoir si une ouverture (ou fermeture) des fenêtres apporte un gain en terme d'efficacité énergétique du bâtiment.

A partir des simulations effectuées, nous pouvons déduire que :

- La distance séparant les noeuds du réseau est un paramètre important à tenir compte lors du déploiement du réseau de capteurs sans-fil. Cette distance doit être minimale pour que chaque noeud consomme le moins d'énergie possible.
- Le nombre de noeuds routeurs (Sink) déployés joue un rôle crucial. En effet, plus grand est le nombre meilleure est la tolérance aux pannes.

2.4.1.3 Modèle Multi-Wall vs. Modèle de FRIIS

Pour une application bâtiment intelligent, les données vont être envoyées d'une pièce à une autre (d'un étage à un autre). Pour cela, un modèle de propagation radio tenant compte les différents obstacles (murs, portes, ...) est requis. Afin d'obtenir des résultats qui s'approchent de la réalité, nous implémentons le modèle Multi-Mur proposé par Motley-Keenan (voir section 2.2.2.3). Dans ce modèle l'onde transmise par un noeud est propagée en suivant le mode espace-libre avec un indice de perte n qui dépend de l'environnement de déploiement du réseau. A la rencontre d'un obstacle une valeur d'atténuation variant selon la matière et l'épaisseur de ce dernier (obstacle) vient se rajouter à la perte du chemin. En vue d'avoir une étude comparative entre les deux modèles Multi-Wall et FRIIS, nous gardons le même cas d'étude Figure 2.14a 2.14b. Grâce à deux kits de développement TI (Texas Instrument) acquis dans le laboratoire, nous relevons les différentes valeurs des atténuations d'obstacles. L'un des kits est configuré comme transmetteur et envoi des données en continu et le deuxième comme récepteur. La technologie utilisée pour la communication est la norme Zigbee avec une fréquence de 2.4 GHz. En utilisant les afficheurs présents sur les deux kits de développement, nous développons un programme d'affichage de la puissance de transmission sur l'un, et de réception sur l'autre Figure 2.19. Nous relevons par la suite les différentes valeurs : transmission et réception. Nous calculons à l'aide de l'équation définissant le modèle Multi-Wall (section 2.3), les deux paramètres : n (facteur d'atténuation), atténuation des murs (L) qui sont comme suit : $n = 2.80$ et $L = 6$.

Path loss exponent (n)	Atténuation des murs (dB)
2.80	6



FIGURE 2.19: Kit de développement Texas Instruments (TI)

La stratégie d'expérimentation est comme suit :

- *Phase 1* : nous commençons d'abord par mettre deux nœuds (kits de développement TI) : un récepteur et un émetteur, sur une ligne droite avec une distance de séparation égale à 1 m. Après cela, les différentes valeurs de réception mesurées sont récoltées. Une moyenne est calculée et nous retenons la valeurs de $FSL = 40dBm$.
- *Phase 2* : Afin de déterminer les valeurs d'atténuation des murs séparant les deux bureaux, nous reprenons la même expérience que la phase 1 avec un mur entre les deux dispositifs (émetteur/récepteur). Vu que les murs sont fabriqués avec la même matière et qu'ils ont les mêmes épaisseurs, nous établissons une moyenne (d'atténuation) des différentes valeurs relevées qui est égale à 6 dB.

Après avoir identifié les valeurs du facteur d'atténuation n et celle de l'atténuation des murs, nous intégrons au programme MATLAB dédié au modèle de connectivité. Ce programme nous fournit une cartographie des valeurs de puissances reçues dans chaque cellule du bâtiment comme le montre la Figure 2.20.

Finalement, nous introduisons ce modèle dans le simulateur réseau NS-2.35 et nous relevons les résultats représentés par les Figure 2.21. L'énergie consommée par un nœud du réseau augmente par rapport à celle consommée en utilisant le modèle FRIIS de propagation radio. Ceci est dû à la prise en compte des obstacles, en effet, chaque nœud va fournir plus d'énergie en transmission pour pouvoir envoyer les messages dans de bonnes conditions. Rajoutons à cela l'effet de retransmission dû aux pertes de paquets durant la communication.

Nous pouvons dire que le modèle Multi-Wall est le plus approprié pour notre application, il sera retenu et exploité pour nos futurs travaux. Ce modèle sera intégré dans la phase de calcul du niveau de connectivité réseau, qui sera détaillée dans le prochain chapitre.

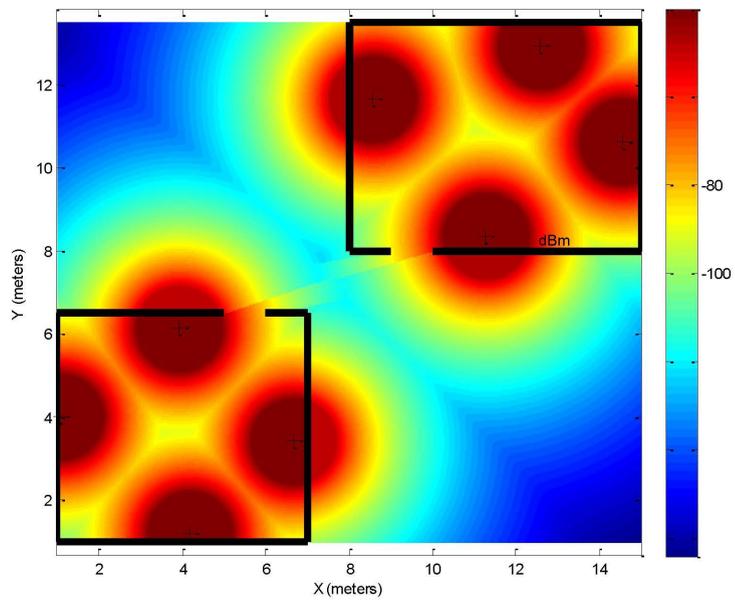


FIGURE 2.20: Puissances de réception dans le bâtiment

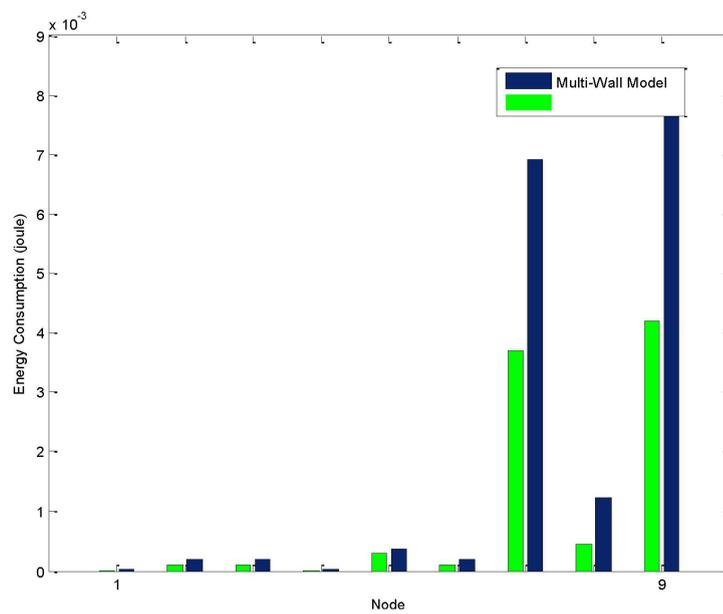


FIGURE 2.21: Énergie consommée par chaque nœud

2.5 Conclusion

Dans ce chapitre nous avons présenté la modélisation des différents critères qui interviennent lors du déploiement d'un RCSF. Nous avons tout d'abord présenté la modélisation de l'espace de déploiement et son inclusion ce dernier dans l'algorithme. Une présentation exhaustive des différents modèles de connectivité et de couverture a été effectuée. Nous avons mis en revu le développement de la technologie radio-communication et comment elle participe au bon fonctionnement du réseau. Pour la couverture des phénomènes physiques (température, présence, ...), trois modèles ont été étudiés dans ce chapitre (modèle binaire, asymptotique et polynomiale). Finalement, Nous terminons la partie modélisation par la présentation du modèle de coût, ainsi que celui de la durée de vie du réseau. Nous pouvons dire que grâce à cette étude approfondie, nous avons toutes les données nécessaires pour pouvoir développer un algorithme d'optimisation capable de prédire les emplacements des différents noeuds constituant le réseau tout en respectant les contraintes calculées par les modèles étudiés.

En se basant sur notre état de l'art, nous avons développé un modèle prenant en compte les quatre contraintes importantes : couverture des phénomènes physiques, connectivité, coût et durée de vie.

Le problème se résume à trouver une topologie du réseau avec le minimum de noeuds possible, tout en assurant une connectivité et une couverture optimale ainsi qu'une durée de vie maximale du réseau. Ce modèle sera présenté dans le chapitre 3 qui sera suivi d'une partie optimisation avec une exposition de la méthode de résolution proposée.

Le chapitre suivant va être dédié aux méthodes de résolution du problème de déploiement des RCSF. Nous commencerons ce chapitre par la définition du problème d'optimisation ainsi qu'une brève introduction sur ce domaine. Après cela, une classification des méthodes selon leurs comportements sera présentée. Les algorithmes d'optimisation sont classés en deux grandes familles : méthodes déterministes (Branch & Bound, simplexe, ...) et méthodes stochastiques ou métaheuristiques (recherche tabou, algorithmes génétiques, ...). Une étude comparative entre ces deux familles ainsi que leurs applications et limites seront exposées. Nous terminons par une comparaison sur l'utilisation de ces méthodes avec celles de l'état de l'art étudié dans le domaine du déploiement des RCSFs.

Chapitre 3

Méthodologie de déploiement du réseau de capteurs sans-fil

3.1 Introduction

Régulièrement, les ingénieurs ainsi que les preneurs de décisions se retrouvent confrontés à des problèmes de complexité grandissante dans différents domaines tels que : systèmes de Production, systèmes de Transports, . . . Dans [107], Michalewicz et Fogel définissent quatre critères pour qu'un problème soit considéré comme complexe :

- Le nombre de solutions possibles dans l'espace de recherche est si grand qu'il est impossible d'effectuer une recherche exhaustive afin de trouver la meilleure.
- Le problème est si compliqué que, pour faciliter la réponse, on doit utiliser des modèles simplifiés du problème qui fournissent des résultats inutiles.
- La fonction d'évaluation qui décrit la qualité de toute solution proposée est perturbée ou varie dans le temps, nécessitant ainsi toute une série de solutions.
- Les solutions possibles sont fortement contraintes de telle sorte que la construction d'une solution réalisable est difficile.

Le problème de déploiement de RCSF est généralement assimilé au problème de galerie d'art (*Art Gallery Problem*) [108–110]. Ce problème se résume à trouver une solution pour le placement d'un nombre minimum de gardes afin qu'ils puissent voir le maximum d'espace dans une galerie d'art (ou musée). Les auteurs dans [111] précisent que : *déterminer le nombre de gardes suffisant afin de couvrir tout l'intérieur d'une galerie d'art à n-murs est un problème NP-Difficile*. Selon les auteurs dans : [112–119] et plus récemment [120–122], le problème de déploiement d'un RCSF dans un immeuble ou bâtiment est similaire au problème de galerie d'art du point de vue où les deux visent

à trouver une combinaison minimale de gardes (ou capteurs) afin de couvrir une galerie d'art ou immeuble. Comme mentionné dans le chapitre précédent un RCSF est repr nous considérerons dans nos travaux que le problème de déploiement des RCSF dans le bâtiment est *NP-Difficile* est nous n'étudierons par la suite que les méthodes de résolutions adaptés. Nous commencerons par introduire un problème d'optimisation et nous présenterons les méthodes de résolution existantes dans la littérature. En s'intéressera par la suite aux méthodes adapté au problème de déploiement des RCSF.

3.2 Résolution du problème de déploiement des RCSF

Selon [123], un problème d'optimisation est défini comme étant une recherche d'une combinaison de paramètres, généralement appelée variables de décision ($x = \{x_1, x_2, x_3, \dots, x_n\}$), qui minimise ou maximise une certaine quantité ordinaire (c) (généralement appelée score ou coût). Cette valeur (c) est assignée par une fonction (f) qui respecte un ensemble de contraintes ($g = \{g_1, g_2, g_3, \dots, g_n\}$) définissant les bornes inférieures/supérieures des variables de décisions (ou bien la faisabilité ou la non-faisabilité de la solution). En général, un cas de minimisation se résume en :

$$\begin{cases} \text{Minimiser} & f(x) \\ \text{sous contraintes} & x \in X \end{cases} \quad (3.1)$$

Afin résoudre ce type de problème plusieurs méthodes ont été proposées dans la littérature [124–126]. Ces méthodes de résolution sont généralement classées selon La fiabilité de la solution finale, le type de variable utilisée, type d'exploration de l'espace de recherche en deux grandes familles : méthodes déterministes, et méthodes stochastiques [123].

3.2.1 Méthodes déterministes

Les méthodes déterministes sont des méthodes qui se caractérisent par une exploration méthodique de l'espace de recherche [127]. Elle procèdent dans un ordre déterminé à l'avance (ex. : direction de la recherche, point de départ (solution initiale) ...).

On retrouve dans la famille des méthodes déterministes les algorithmes : simplexe [128, 129], Quasi-newton [123], Gradient [129], Branch & Bound [130]. Nous allons présenter une brève introduction des trois méthodes citées.

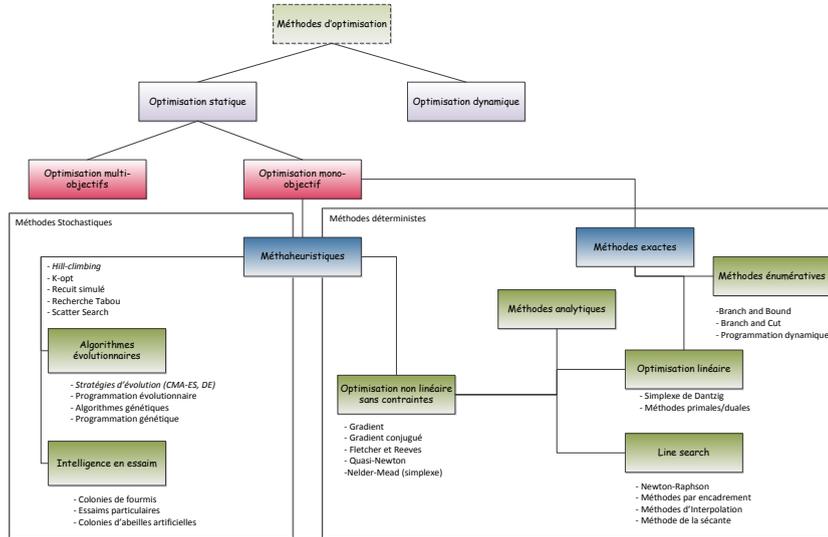


FIGURE 3.1: Classification des différentes méthodes d'optimisation [11]

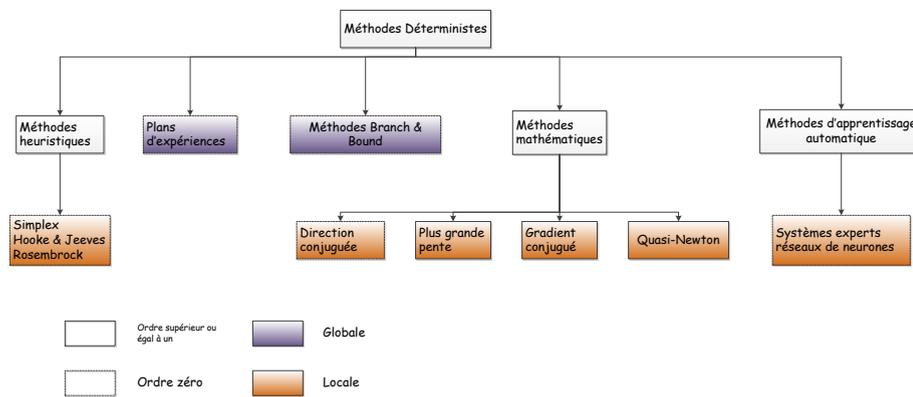


FIGURE 3.2: Classification des méthodes déterministes[12]

3.2.1.1 Méthode de gradient

Ces méthodes sont les plus anciennes. Elles permettent de résoudre un problème non-linéaire[131] et sont basées sur la connaissance de la dérivée de la fonction objectif en chacun des points de l'espace de recherche. En règle générale, le fonctionnement de la méthode commence par un choix d'une solution de départ x_0 qu'il veut améliorer, en plongeant vers l'optimum le plus proche. Pour cela, l'algorithme calcule $\nabla f(x_0)$ le gradient de la fonction objectif au point x_0 . Après cela, il obtient la direction de la plus grande pente $-\nabla f(x_0)$, sur laquelle il va avancer d'un pas α_k . La solution est ensuite remplacée par $x_{k+1} = x_k - \alpha \nabla f(x_k)$.

3.2.1.2 Méthode du simplexe

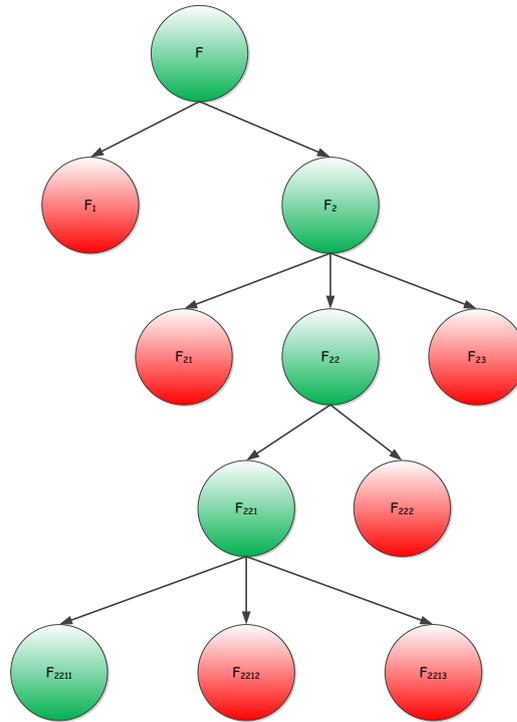
Appelée aussi méthode de Nelder-Mead [132], cette méthode est la plus courante des méthodes d'optimisation locales. C'est une méthode directe à solutions multiples, où l'algorithme n'améliore pas une seule solution pendant une itération, mais un ensemble de solutions appelé "polytope". Le principe de fonctionnement de cette méthode est comme suit : (1) Une solution est choisie au hasard (par tirage aléatoire) dans l'espace de recherche (x_1), (2) le reste du polytope est généré grâce à la fonction $x_{i+1} = x_1 + \lambda e_i, \forall i \in [1, n]$ où les e_i sont des vecteurs unitaires linéairement indépendants et les λ une constante adaptée à la caractéristique du problème, (3) À chaque itération les points x_i sont ordonnés de manière à avoir $f(x_1) \leq f(x_2) \leq f(x_3) \leq \dots \leq f(x_{n+1})$ dans le cas d'une minimisation. Les nouvelles solutions sont ensuite générées en utilisant des notions de transformations géométriques élémentaires (réflexion, contraction, expansion, et rétrécissement). A chaque itération la mauvaise solution est remplacée par la nouvelle calculée, jusqu'à ce que la différence entre les deux (la pire et la meilleure solution) soit inférieure à un seuil prédéfini. L'intérêt principale de cette méthode réside dans la non-nécessité de calcul du gradient. Cette méthode se base uniquement sur l'évaluation de la fonction objectif, ce qui la rend utilisable pour des fonctions bruitées [11].

3.2.1.3 Branch & Bound

L'algorithme consiste à diviser le problème initial F en plusieurs sous-problèmes (sous-ensembles) F_1, F_2, \dots, F_n , cela est résumé dans le terme *Branch*. Si les sous-problèmes sont aussi difficiles que le problème original, ces derniers sont partitionnés en sous-problèmes eux aussi (Figure 3.3). On suppose dans cet algorithme que l'on peut calculer une borne inférieure $b(F_i)$ sur le coût optimal qui existe dans un sous-ensemble F_i (Bound). À chaque instant, l'algorithme maintient : une liste de sous-problèmes actifs, un coût U de la meilleure solution obtenue jusque-là, la valeur initiale de U . Les différentes étapes de l'algorithme sont : (1) sélectionner le coût initial U (depuis une solution initiale praticable), (2) diviser F en plusieurs sous-problèmes, (3) sélectionner un sous-problème actif F_i , (4) si F_i est non admissible, le supprimer, sinon, calculer $b(F_i)$, (5) Si $b(F_i) \geq U$, supprimer F_i ; (6) Sinon soit résoudre F_i directement soit recommencer de (2)[133].

3.2.1.4 Branch & Cut

L'algorithme de *Branch & Cut* est un *Branch & Bound* classique auquel on rajoute une génération dynamique de contraintes, appelées *coupes*, car elle servent à couper le point

FIGURE 3.3: Division du problème F en sous-problèmes

fractionnaire courant[134]. La méthode de coupe est appliquée localement sur chaque nœud et de manière répétitive jusqu'à l'obtention d'une solution entière optimale. Cette méthode s'est avérée très efficace pour la résolution de problèmes d'optimisation combinatoires difficiles (ex. : TSP, coupe maximum, ...) [135]. Malgré les progrès réalisés dans le domaine des méthodes d'optimisation exactes, elles restent des méthodes appliquées à des problèmes de petite taille (leurs temps de calcul risquent d'augmenter exponentiellement avec la taille du problème). Les méthodes approchées (stochastiques) sont une alternative intéressante pour trouver des solutions s'approchant de l'optimale dans des délais plus courts. Certes, la solution trouvée n'est pas toujours optimale mais le temps de calcul nécessaire afin de la trouver est considérablement minimisé. Nous allons présenter dans la section suivante quelques méthodes qui ont connues un grand succès dans ce domaine.

3.2.2 Méta-heuristiques

Les méta-heuristiques sont la catégorie la plus importante des méthodes stochastiques qui s'appuient sur des notions de probabilité et des mécanismes aléatoires pour trouver la solution optimale. D'après cette définition, on peut dire qu'une succession d'exécutions de ces méthodes peut générer des solutions différentes pour une même configuration

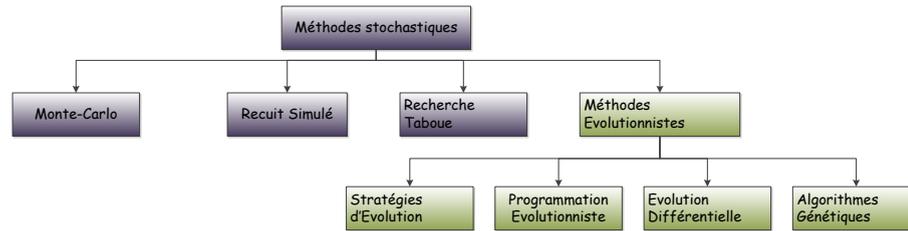


FIGURE 3.4: Exemple de méthodes stochastiques[12]

initiale du problème d'optimisation. La plupart des algorithmes issus des domaines de l'intelligence Artificielle (Computational Intelligence), l'informatique bio-inspirée et les méta-heuristiques, sont des méthodes stochastiques. Ces algorithmes effectuent un échantillonnage aléatoire de l'espace de recherche d'une manière partielle en se concentrant sur les domaines d'intérêt et négligeant les zones moins intéressantes [136].

Parmi les points forts de ces méthodes on peut citer la capacité à converger (généralement) vers l'optimum global, et cela, sans une grande influence du choix de la solution initiale ou point de départ. Elles sont généralement d'ordre zéro. Cependant, un nombre important de générations (itérations) de ces algorithmes, est parfois nécessaire pour trouver une solution dans le voisinage de l'optimum global [12]. La Figure 3.4, énumère quelques méthodes appartenant à cette catégorie. Les méta-heuristiques peuvent être considérés comme un cadre algorithmique général qui peut être appliqué à différents problèmes d'optimisation avec quelques modifications relatives pour les adapter à un problème spécifique [137, 138]. Il existe plusieurs méthodes méta-heuristiques tels que : recherche locale, recherche Tabou, Algorithmes Génétiques, algorithmes de fourmis, recuit simulé. Blum et Roli décrivent neuf propriétés des méta-heuristiques[139], qui sont comme suit :

- Les méta-heuristiques sont des stratégies qui guident le processus de recherche,
- Le but est d'explorer efficacement l'espace de recherche afin de trouver des solutions quasi-optimales,
- Les techniques qui constituent les méta-heuristiques vont du plus simple "recherche locale" jusqu'à des processus d'apprentissage plus complexes,
- Les algorithmes méta-heuristiques sont généralement approximatifs (en terme de solution) et non-déterministes (n'assurent pas l'optimalité de la solution),
- Elles peuvent inclure un mécanisme pour éviter d'être pris au piège dans des espaces confinés de l'espace de recherche
- Les concepts de base de méta-heuristiques permettent une description de niveau abstrait
- Les méta-heuristiques ne sont pas spécifiques à un problème
- Les méta-heuristiques peuvent faire usage des connaissances spécifiques à un domaine sous la forme d'heuristiques qui sont contrôlés par la stratégie de niveau supérieur

- Les méta-heuristiques utilisent des notions d'expérience de recherche (en forme de mémoire) pour guider la recherche

Ces algorithmes sont généralement appliqués à des problèmes complexes (dont la complexité est NP-Complexe, ou NP-Difficile), pour lesquels des approches traditionnelles ne sont pas adaptées[140]. Les méta-heuristiques peuvent être classées en deux grandes familles : méta-heuristiques basées sur une seule solution (*S-métaheuristique*) et méta-heuristiques basées sur une multitude de solutions regroupées dans un ensemble appelé population (*P-méta-heuristiques*). Dans la première catégorie, les méthodes utilisées manipulent une seule solution dans chaque itération. Leur plus grand avantage est leur rapidité d'exécution, par contre elles sont connues pour leur convergence prématurée. Dans la seconde, une nouvelle population de solutions est créée. Ceci favorise l'exploration de l'espace de recherche, ce qui permet de trouver les meilleures solutions. Par contre, ces algorithmes ont tendance à consommer énormément de mémoire quand ils sont appliqués à des problèmes complexes de très grande taille[141]. Dans les sections suivantes, nous allons détailler quelques exemples de méta-heuristiques : recherche Tabou, Recuit simulé, algorithmes évolutionnaires et intelligence en essaim.

3.2.2.1 Recherche Tabou

Proposée par Fred Glover en 1986 [142], son idée principale consiste à garder en mémoire l'historique des solutions déjà explorées. L'algorithme commence par créer et initialiser une liste vide L de taille fixe. Cette liste contiendra toutes les solutions explorées et qui deviennent interdites ou "*Tabou*". Ceci permet à l'algorithme de ne pas tomber dans des cycles répétitifs (comme c'est le cas avec la méthode de recherche locale) et d'éviter ainsi les minimas locaux. Le choix de la solution va s'effectuer de manière à ce que l'on choisisse la meilleure solution dans le voisinage, en évitant de choisir ceux contenus dans la liste Tabou.

3.2.2.2 Recuit Simulé

Cette méthode tire son nom de la physique des matériaux et plus précisément de la métallurgie. Le "*Recuit*" est une opération qui consiste à laisser refroidir lentement un cristal en vue de l'amener à sa structure d'énergie minimale. Un métal (ou cristal) qui est refroidi brutalement peut perdre ses capacités de résistivité et peut être bloqué dans un état peu favorable. Alors qu'au contraire, un refroidissement lent permettra aux molécules de se stabiliser et de rester dans une configuration stable. Le principe de l'algorithme consiste à imiter (simuler) la méthode du recuit pour pouvoir trouver une solution quasi-optimale, tout en évitant de tomber dans des minimas locaux [129].

Cette méthode peut être vue comme une généralisation (ou amélioration) de la méthode de descente stochastique. Elle répète une procédure itérative qui cherche à minimiser une fonction tout en acceptant, de manière contrôlée, des solutions qui la dégrade (la fonction). Dans chaque itération, une solution (s') dans le voisinage de la solution actuelle (s) est générée de manière aléatoire. Cette nouvelle solution sera soit retenue pour remplacer s , soit rejetée. Si la performance de cette solution voisine est meilleure que celle de la solution actuelle, la solution est automatiquement retenue ($f(s') \leq f(s)$). Dans le cas contraire, la solution est retenue avec une probabilité $p(\Delta f, T)$, qui dépend de deux facteurs : $\Delta f = f(s') - f(s)$, et de T qui est la température (une température plus élevée donne une probabilité d'acceptation plus grande).

3.2.2.3 Algorithmes Évolutionnaires

Les phénomènes physiques et biologiques ont inspiré un bon nombre d'algorithmes dans le domaine de l'optimisation combinatoire. Une multitude d'algorithmes bio-inspirés ont montré leur robustesse face à des problèmes complexes. Les algorithmes évolutionnaires (EA) sont l'un des exemples d'algorithmes bio-inspirés qui peuvent traiter des problèmes NP-Difficile. Les EAs sont basés sur la théorie de l'évolution proposée par Darwin en 1807 [143]. L'idée principale consiste à ce que l'apparition des espèces adaptées est une conséquence de deux phénomènes principaux : (1) la sélection naturelle (les individus les plus adaptés survivent et se reproduisent), (2) plusieurs variations peuvent se produire sur le matériel génétique des espèces. En termes d'optimisation, l'évolution se traduit par un processus itératif de recherche de l'optimum dans l'espace de recherche.

D'après [13] on peut dire qu'il existe une multitude de variante au sein des algorithmes évolutionnaires, qui se sont développées de manière indépendante : La Programmation Évolutionnaire (EP) parue dans les années 60 [144], les Algorithmes Génétiques (GA) introduits par J. Holland [145] en 1975, ces derniers ont été promus par son élève D. E. Goldberg [146] dix ans plus tard (1989), Les Stratégies d'Evolution (SE) [147], et finalement la programmation génétique proposée par J.Koza dans ces deux articles [148, 149].

Soit F une fonction à optimiser (minimiser-maximiser), définie sur un espace Ω . Dans un vocabulaire d'EA on notera :

- La fonction F est appelée fonction de fitness ou fonction d'adaptation. Chaque individu dispose de sa propre valeur de fitness ou d'adaptation.
- Les points de l'espace de recherche sont appelés individus et peuvent être représentés par un ou plusieurs chromosomes.
- Un chromosome représente une solution ; il peut être codé par des valeurs binaires, réelles,...

- Un ensemble P d'individus est appelé population
- Chaque itération de l'algorithme produit un nouvel ensemble de solutions appelé nouvelle génération
- Les individus de la N^{ieme} génération sont appelés parents et ceux de la $(N + 1)^{ieme}$ génération sont appelés enfants
- À chaque génération les parents sont recombinaés pour générer de nouveaux enfants (nouvelles solutions)

L'architecture typique d'un algorithme évolutionnaire est représentée dans la Figure 3.5. Nous pouvons distinguer les étapes les plus importantes de l'algorithme citées auparavant (sélection, reproduction, évaluation). D'une manière générale, la population initiale (solutions de départ) est générée de manière aléatoire. Une fois ces individus générés, ils passent par une multitude d'opérations (sélection et reproduction) afin de trouver une solution à un problème donné. Les principales étapes d'un EA, à savoir : sélection, reproduction, évaluation, sont répétées jusqu'à l'atteinte d'un critère d'arrêt (ex. : nombre maximum de générations).

3.2.2.4 Intelligence en Essaim

D'après [150] l'optimisation par Essaim Particulaire (OEP, ou PSO en Anglais) s'inspire du comportement social des animaux évoluant en essaim tel que les bancs de poissons par exemple. Cette méthode a été proposée par Kennedy et Eberhart en 1995 [151]. Au départ, les deux chercheurs voulaient simuler la capacité des oiseaux à voler d'une façon synchrone et leur aptitude à changer brusquement de direction tout en gardant en une formation optimale. Le modèle qu'ils ont proposé a ensuite été étendu en un algorithme simple et efficace d'optimisation [152]. On observe dans ce type d'organisme un déplacement relativement complexe du groupe, alors que, chaque individu dispose d'une connaissance strictement locale de sa situation dans l'essaim, et a une intelligence très limitée [153]. Grâce à une connaissance de sa propre position dans l'essaim et des informations sur la vitesse et la position de ses voisins, un individu peut décider de son propre déplacement. Ces déplacements s'effectuent en employant des règles simples telles que : "aller à la même vitesse que les autres", "se déplacer dans la même direction" ou encore "rester proche de ses voisins", qui maintiennent la cohésion de l'essaim, et permettent la mise en œuvre de comportements collectifs complexes et adaptatifs. Ce comportement collectif propre à ce type d'animaux qui vivent en essaims, et qui se base sur l'analyse de l'environnement et du voisinage pour effectuer des choix optimaux, constitue alors, une méthode d'optimisation par l'observation des tendances des individus voisins. Chaque individu cherche à optimiser ses chances en suivant une tendance qu'il modère par ses propres vécus [154].

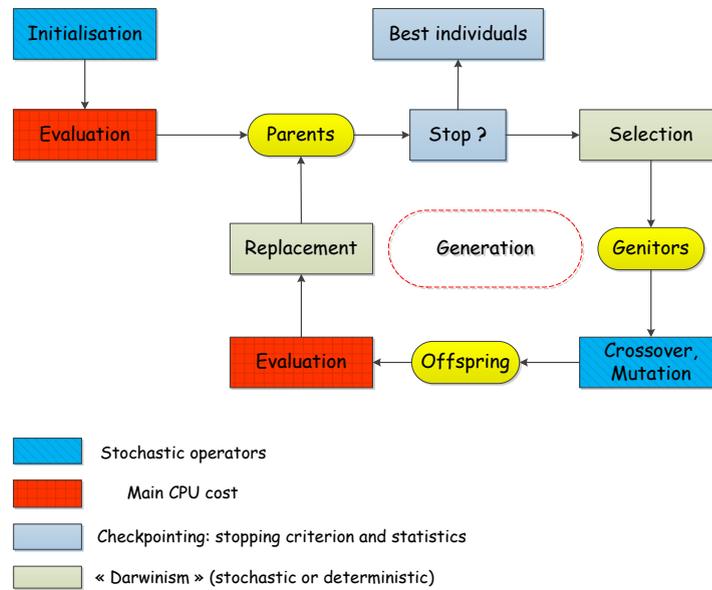


FIGURE 3.5: Architecture d'un algorithme évolutionnaire [13]

3.2.3 L'optimisation dans le problème de déploiement des RCSFs

Comme on l'a mentionné auparavant, le problème de déploiement des RCSFs est un problème NP-Difficile. Par conséquent, ce problème ne pourrait pas être résolu grâce à des méthodes exactes, mais résolu par des méthodes non-conventionnelles telles que les algorithmes évolutionnaires [155]. Ce problème peut être décrit comme suit

Soit A un espace à surveiller, dans lequel un RCSF va être déployé. Chaque capteur ou routeur (station de base ou sink) dispose d'un rayon de communication r_{comm} et chaque nœud capteur dispose d'un rayon de captage r_{cov} . Un capteur (C) déployé, dans la position (l, p) doit assurer deux tâches essentielles : collecter des données sur les phénomènes physiques se produisant dans toutes les parties du bâtiment (positions (i, j)), et les communiquer à la plus proche station de base. Ceci n'est possible que si la distance euclidienne, séparant la position de l'évènement et le capteur, est inférieure au rayon de couverture r_{cov} ($(l - i)^2 + (p - j)^2 \leq r_{cov}^2$). Les informations ainsi collectées doivent être transmises au routeur le plus proche. Soit un routeur (station de base ou sink) localisé à la position (k, t) . Pour que la communication entre ce routeur (R) et le nœud capteur C soit établie, il faut que la puissance du signal transmis par l'un et reçu par l'autre soit supérieure à un seuil prédéfini par le constructeur, appelé sensibilité de l'antenne de réception.

L'objectif de notre travail est de trouver un compromis entre la minimisation du nombre de nœuds formant le réseau, et la qualité de service du réseau déployé. Ceci n'est possible que si l'on utilise des méthodes de résolution multi-objectifs. Par la suite, nous

Year	First Author	Objective(s)	Used Method(s)	Considered Node(s)	Indoor/Outdoor
2007	K. P. FERENTINOS	- Coverage - Lifetime	- Classical GA	- Agricultural Sensor	- Outdoor
2008	A. PINTO	-Connectivity	- ILP	Relay Nodes	-Indoor
2009	N. AITSAADI	-Cost - Quality of Monitoring	- Tabu Search	-Sensor Nodes	-Outdoor
2010	X. M. HU	-Coverage -Lifetime	-Hybrid GA	- Sensor Nodes	- Outdoor
2010	E. GUNEY	- Lifetime - Routing Protocol	- ILP	-Sink Nodes	- Outdoor
2010	A. KONSTANTINIDIS	- Coverage - Lifetime	- MOEA/D	- Sensors with multiple transmission power levels	- Outdoor
2011	J. A. PEREZ	- Lifetime - Connectivity	- MEMETIC Algorithms	- Relay Nodes	- Outdoor
2011	N. AITSAADI	- Cost - Connectivity - Lifetime - Quality of Monitoring	- Tabu Search - Potential Field Approach	- Sensors for geographical events monitoring	- Outdoor
2011	X. XU	- Lifetime	- Heuristic	- Relay Nodes	- Outdoor
2012	J. Y. LEE	- Cost - Coverage	- MOGA	- RFID Sensors	- Indoor
2013	Y. YOON	- Coverage	- Classical GA	- Sensors with different sensing range	- Outdoor
2013	L. SITANAYAH	- Cost - Fault Tolerance	- GRASP	- Sink and Relay Nodes	- Outdoor
2013	S. ÖZDEMIR	- Lifetime - Coverage - Cluster-head choice	- MOEA/D - NSGA-II	- Sensors	-Outdoor
2014	D. HE	- Cost - Connectivity - Coverage - Lifetime	- MOGA	- Sensor and Relay nodes for building application	- Indoor

TABLE 3.1: rasumé des travaux qui ont utilisé de l'optimisation Multi-Objectif des RCSFs

allons présenter un état de l'art sur les travaux effectués dans cette optique. Le tableau (Tab. 3.1) résume les principaux travaux de recherche qui optimisent le placement des capteurs sans fil avec des méthodes mono ou multi-objectifs).

J. Y. Lee et al., 2012 [156], ont traité le problème de placement des nœuds RFID dans un environnement indoor. Le but du travail était de placer un nombre minimal de tags RFID, tout en assurant une couverture totale de la RI (Région d'Intérêt). Les auteurs ont proposé un algorithme d'optimisation hybride multi-objective nommée MOASA (Multi-Objective Algorithm for Sensor Arrangement). La méthode hybride consiste à utiliser deux méthodes d'optimisation : mono-objective, et multi-objective. Dans un premier temps, l'algorithme commence par chercher l'emplacement des tags RFID avec pour objectif la maximisation de la couverture de l'espace de déploiement, jusqu'à atteindre un certain seuil prédéfini. Ensuite, les trois principaux objectifs (couverture, sur-couverture, coût) sont regroupés dans deux fonctions objectifs (F_1 : couverture et coût, F_2 : coût et

sur-couverture), qui sont résolus par la suite par un algorithme multi-objectifs d'optimisation. Les résultats numériques montrent la performance de la méthodologie d'optimisation utilisée, qui est comparée par la suite, avec des résultats obtenus par d'autres méthodes évolutionnaires (AG, DE, et SPEA-II).

K. P. Ferentinos et al. 2007, [157], ont proposé une méthode d'approximation pour résoudre le problème de placement des nœuds capteurs. Le but étant de trouver le nombre optimal de nœuds affectés à un groupement (cluster), afin de maximiser la durée de vie du réseau ainsi que la couverture des nœuds. Le système a été formulé par un problème d'optimisation multi-objectif et résolu par un algorithme génétique. Afin de minimiser la complexité de l'algorithme, le nombre de nœuds à déployer est connu auparavant, donc, pas d'optimisation au niveau du nombre de nœuds à déployer qui ont un impact sur le coût. Nous constatons aussi, que l'algorithme nécessite 3000 générations pour fournir une solution, et que la solution est représentée par 1000 variables, ce qui cause une grande consommation en terme de temps de calcul et de mémoire.

A. Konstantinidis et al. 2010 [158], ont proposé une méthode basée sur les algorithmes évolutionnaires multi-objectif, pour l'optimisation du placement des nœuds capteurs sous les contraintes de durée de vie et de couverture. Les auteurs ont pris en considération des capteurs disposant d'une multitude de niveaux de puissance de transmission. Le but de la méthode était de trouver la meilleure disposition des nœuds du réseau (coordonnées), ainsi que leur niveaux de transmission (puissance), afin d'avoir une couverture et une durée de vie optimale du réseau de capteurs sans-fil.

Le problème de placement des capteurs sans-fil disposant de plusieurs niveaux de détection a été traité dans Y. Yoon et a. 2013 [159]. En se basant sur un ensemble de capteurs sans-fil connus (nombre, caractéristiques, . . .), l'algorithme proposé essaie de trouver une solution optimale au problème de placement sous la contrainte de couverture. Les auteurs ont utilisé l'algorithme génétique mono-objectif, avec une méthode de croisement BLX- α et une mutation gaussienne, pour résoudre le problème. Pour l'évaluation de la solution, les auteurs ont utilisé une méthode de Monte-Carlo.

HU, et al. 2010 [160] proposent une méthode pour traiter le problème : Maximum Disjoint Set Covers Problem (MDSCP) afin d'étendre la durée de vie du RCSF. Le problème se résume à trouver le maximum de couverture par ensemble (disjoint set covers), où chaque ensemble peut couvrir le maximum de cibles (d'espace). Avec cette définition, la durée de vie du réseau est définie par le nombre d'ensembles disjoints. Ceci est réalisable, car chaque ensemble couvrant est activé pendant une unité de temps (les ensembles restant sont désactivés pendant cette période). Le problème est connu pour être NP-Difficile [156]. Pour cela, les auteurs ont utilisé un algorithme génétique hybride, qui inclut une méthode d'ordonnancement des opérations de transition (activation des ensembles). L'algorithme proposé a été comparé à d'autres algorithmes issus de la littérature : Genetic

Algorithms for Maximum Disjoint Covers (GAMDSC) (C.C. Lai et al. 2007), et l'heuristique Most Constrained-Minimum Constraining (MCMCC) (S. Slijepcevic and M. Potkonjak, 2001).

Une méthode pseudo-aléatoire, basée sur l'heuristique recherche Tabou, a été utilisée dans [90] par N. AITSAADI, et al. 2009, pour déterminer le meilleur nombre de nœuds, ainsi que leurs positions optimales, afin de couvrir en totalité la région surveillée. Les auteurs ont considéré un modèle de détection probabiliste des nœuds capteurs, qui est plus réaliste qu'un modèle binaire. Les résultats obtenus ont montré que la méthode proposée surpasse plusieurs autres approches proposées dans la littérature, comme le déploiement aléatoire, par exemple. Ce travail a été étendu dans [82], pour inclure les contraintes de durée de vie et de connectivité. L'approche potentielle de champ (potential field approach), qui est très connue dans le domaine de la robotique, a été utilisée dans ce travail avec la recherche Tabou.

La plupart des travaux ci-dessus se focalisent sur le positionnement des capteurs sous différentes contraintes, en traitant les objectifs individuellement (un seul objectif, voire deux au plus) pour la plupart. Les nœuds sink ou relais n'ont pas été pris en compte par la majorité de ces travaux. Un déploiement efficace d'un RCSF, doit inclure tous les nœuds du réseau : plusieurs nœuds capteurs, et plusieurs nœuds relais (sink). La définition des positions des nœuds sink/relais est une problématique importante et ne peut pas être ignorée lors du déploiement du RCSF. Par conséquent, quelques travaux traitant ce problème (plusieurs objectifs à la fois et prise en compte des nœuds sink) sont étudiés ci-dessous.

J.A. Perrez et al. 2011, [161], traitent le problème de placement des nœuds relais pour l'extension de la durée de vie du RCSF. Basée sur un ensemble de capteurs déjà déployés, les auteurs proposent une méthode qui calcule le nombre des nœuds relais, ainsi que leur positionnement nécessaire afin d'augmenter l'efficacité énergétique du réseau. Une méthode hybride, incluant un algorithme génétique et une méthode de recherche locale, a été utilisée pour la résolution du problème. La fonction d'évaluation (fitness) a été donnée par une somme pondérée des différents objectifs. E. Guney et al, 2010, [162], ont formulé le problème en un ensemble d'équations mathématiques, afin de les résoudre en utilisant la programmation linéaire, pour le problème de déploiement et routage des nœuds sink (Joint Sink Location and Routing Problem (SLRP)), en considérant une multitude de nœuds sink stationnaires. E. Guney et al, 2010, [162], ont utilisé la programmation linéaire afin de résoudre le problème de déploiement des nœuds relais (routeur ou sink). Le problème étudié se résume à trouver les positions optimales de ces nœuds dans un bâtiment. Le nombre de nœuds à déployer est connu a posteriori. Par conséquent, il n'y a pas d'optimisation par rapport au nombre de nœuds qui intervient dans le calcul du coût de déploiement.

L'algorithme proposé définit les positions optimales, ainsi que l'algorithme de routage adopté, des nœuds sink afin de maximiser la durée de vie du RCSF. Les auteurs ont proposé deux principaux modèles de programmation en nombre entiers, pour résoudre le problème de déploiement dans les RCSF. Dans le premier, le nombre des nœuds sink est connu dès le début, et l'algorithme cherche les positions optimales, afin de minimiser l'énergie totale consommée par le routage. Dans le second, la position ainsi que le nombre de nœuds sink est inconnue. La programmation linéaire en entiers a été utilisée par Pinto, et al. 2008 [163] pour résoudre le problème de placement des nœuds relais pour une application indoor. Le problème était de trouver les coordonnées optimales d'un ensemble de nœuds relais (nombre connu) pour assurer une bonne connectivité du réseau. Ce type de méthode d'optimisation assure l'exactitude de la solution fournie. A notre connaissance, seule une optimisation mono-objectif avec un nombre connu de nœuds et des placements connus et peu nombreux peut être résolu avec une méthode exacte telle que la programmation linéaire en entiers. Avec l'introduction de la fonction coût (nombre inconnu de nœuds), l'espace de recherche devient très grand, ce qui provoque une très forte consommation du temps de calcul et de la mémoire.

Les travaux cités ci-dessus, souffrent de plusieurs limites. Certains travaux ont traité le problème de déploiement en considérant un seul type de nœuds (capteurs ou routeurs), ce qui ne représente pas un déploiement exhaustif du réseau. Aussi, la plupart de ces travaux partent du principe que le nombre de nœuds est fixe et connu a posteriori, donc le coût ne peut pas être optimisé. En dépit de quelques travaux (20%) réalisés sur des applications indoor, le reste traite le déploiement des RCSF dans des environnements outdoor. Sans une considération exhaustive des contraintes liées au déploiement d'un RCSF (coût, couverture, et connectivité), les performances du réseau ne peuvent être totalement optimisées.

3.3 Architecture proposée

La phase de conception d'un RCSF consiste à déterminer la topologie du réseau (nombre de nœuds nécessaires et leurs positions respectives), en respectant des valeurs optimales des différents objectifs précédemment cités. Cette tâche peut être exprimée par un problème d'optimisation multi-objectifs qui a été prouvé comme étant NP-Difficile [88]. Afin de résoudre un tel problème nous avons retenu cinq algorithmes évolutionnaires : Genetic Algorithms (GA), Non-Dominated Sorting Genetic Algorithm (NSGA-II)[164], Strength Pareto Evolutionary Algorithm (SPEA-II)[155, 165], Pareto Envelope based Selection (PESA-II)[166], et Differential Evolution (DE)[167, 168]. Ces algorithmes sont, pour le meilleur de notre connaissance, très utilisés pour trouver des solutions proches de

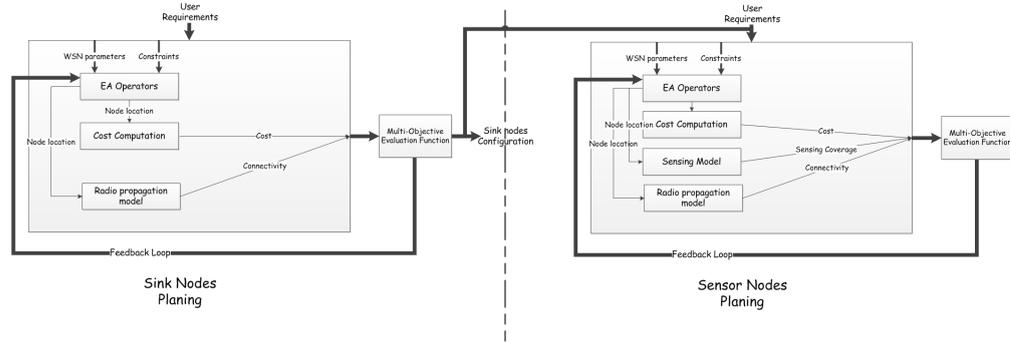


FIGURE 3.6: Architecture du système de déploiement du RCSF

l'optimal dans un délai minimal lors de la résolution de problèmes NP-Difficile comme le problème de déploiement des RCSF. De plus, le déploiement des RCSF est un problème multi-objectifs NP-difficile, ce qui valide le choix des méthodes sélectionnées. Ces algorithmes sont caractérisés par trois phases distinctes, qui sont : (1) Sélection et évaluation, (2) Reproduction, (3) Classement des solution et remplacement. Nous considérons dans notre travail une architecture à deux niveaux composée comme suit :

- *Déploiement de routeurs (sinks)* : dans ce niveau, seuls les deux objectifs coût et connectivité sont pris en compte . L'objectif est d'avoir un nombre minimal de nœuds routeurs à déployer, tout en satisfaisant un $RSSI$ supérieur ou égale à RX_{Tsh} . Dans cette partie seuls les nœuds routeurs sont déployés.
- *Déploiement de nœuds capteurs* : En se basant sur la solution obtenue dans la première phase, l'algorithme est ré-exécuté pour avoir les positions , ainsi que le nombre optimal des nœuds capteurs . Dans cette partie de l'algorithme, on considère quatre fonctions objectifs : coût, connectivité, durée de vie et couverture des évènements.

Cette architecture est représentée dans la Figure 3.6. Nous pouvons distinguer que nous avons choisi de commencer par l'optimisation du nombre et position des nœuds routeur (sink). Ceci est du à leurs capacité d'interconnerter les différents autres nœuds capteurs, en vue de collecter les données. Aussi, ceci nous offre la possibilité d'étendre le réseau dans le futur. En effet, l'un des objectifs de déploiement des nœuds routeurs est la connectivité. Chaque partie du bâtiment doit être couverte par le signal radio des routeurs déployés. Ceci nous offre la possibilité de positionner un nœud capteur à n'importe quel endroit du bâtiment, il sera automatiquement relié au routeur couvrant cette zone. Un autre avantage de ce découpage est que l'on peut l'utiliser dans tous les cas suivant :

- Déploiement dans un bâtiment en cours de construction,
- Déploiement dans un bâtiment existant (construit) et ne disposant pas de RCSF,
- Déploiement de routeurs (bâtiment déjà équipé),
- Déploiement de capteurs (bâtiment déjà équipé).

3.3.1 Méthodes utilisées

3.3.1.1 Algorithme génétique (AG)

L'algorithme génétique est une stratégie adaptative et méthode d'optimisation globale. Cet algorithme s'inspire de la génétique de population, ainsi que, la compréhension de la structure mendélienne (chromosome, gènes, allèles, ...) et ses mécanismes (recombinaison et mutation)[140]. Cet algorithme opère en quatre parties distinctes : Sélection, Croisement, Mutation, et Évaluation. Dans chaque génération de l'algorithme, les parents sont sélectionnés pour entreprendre plusieurs opérations (croisement et mutation). Chaque opération est réalisée avec une probabilité pré-définie.

La phase de sélection détermine le nombre de fois qu'un individu participe à la reproduction en une seule génération de l'algorithme. Les individus ayant les meilleures valeurs de *fitness* seront fréquemment sélectionnés par rapport aux autres. Les individus sélectionnés sont appelés parents, et sont choisis pour participer à la phase de reproduction. Nous adoptons une sélection par tournoi, pour choisir les individus participants à la phase de reproduction. Le croisement consiste à l'application d'opérations avec une certaine probabilité P_c aux individus précédemment sélectionnés. La probabilité de croisement définit la proportion de la population (nombre de parents) qui va être utilisée par un opérateur de croisement. Dans les premières générations de l'algorithme, l'opérateur de croisement joue un rôle d'exploration de l'espace de recherche[169]. Ceci est vrai, car l'opération de croisement génère de nouvelles solutions qui se trouvent dans des espaces inexplorés auparavant. Les nouvelles solutions générées par les opérations de reproduction et évaluées, sont ensuite classées et une phase de remplacement de la population est entreprise. Dans cette phase, les meilleures solutions sont sélectionnées afin de remplacer la population de la génération précédente.

Il existe une multitude de méthodes de croisement qui sont adaptées à des problèmes spécifiques. À partir des travaux effectués dans cette section, nous pouvons dire qu'il existe 21 types de croisement. Ces types de croisement diffèrent dans la manière de découpage et d'assemblage des chromosomes (parents) pour trouver de nouvelles solutions (enfants). Nous utilisons dans notre algorithme un croisement à un et deux points. Ce type de croisement sera expliqué en détails dans les sections suivantes.

Une autre opération de reproduction est l'étape de mutation avec une plus petite probabilité notée P_m appelée taux de mutation. Cet opérateur introduit une perturbation à la solution, ce qui maintient une diversité au sein de la population. L'opérateur de mutation évite la convergence prématurée à un optimum local. Nous intégrons dans notre algorithme une méthode de mutation binaire qui sera détaillée dans les sections suivantes.

3.3.1.2 SPEA-II

Strength Pareto Evolutionary Algorithm (SPEA) est un algorithme évolutionnaire, qui appartient à la famille des algorithmes évolutionnaire multi-objectifs (EMO). Cet algorithme représente une version étendue de l'algorithme génétique qui existe en deux versions : SPEA et son extension améliorée SPEA-II[140].

La première version de cet algorithme a été proposée par E. Zitzler et al. [155, 165]. Les nouveautés introduites dans cet algorithme par rapport à l'algorithme génétique classique sont comme suit :

- Une copie des solutions non-dominées (jusque là) appelée *Archive* est stockée à chaque génération de l'algorithme,
- Utilisation de la notion de dominance selon *Pareto* pour l'affectation des valeurs de *fitness* pour chaque individu,
- Utilisation d'une méthode de clustering (regroupement) similaire à l'algorithme des *k*-plus proches voisins pour réduire le nombre des individus dans la base de données *Archive*,

Les étapes suivantes sont réalisées pendant chaque génération de l'algorithme :

- Deux ensembles d'individus sont créés : (1) population initiale, et (2) *Archive* vide,
- Toute solution (individu) non-dominée est stockée dans l'*Archive*. Si la taille des solutions non-dominées est très grande (ou des doublons existent dans l'*Archive*), les anciens membres contenus dans l'*Archive* sont supprimés en utilisant une technique de clustering qui préserve les caractéristiques du front de Pareto (solutions non-dominées),
- Une valeur de puissance $S(i) \in [0, 1[$ est assignée à chaque membre i de l'ensemble *Archive*. $S(i)$ représente le pourcentage d'individus (contenu dans la population courante) qui sont dominés par la solution i ,
- Une valeur de *fitness* $f(j)$ est assignée à l'individu j de la population. Cette valeur est calculée par la somme de toutes les valeurs $S(i)$ de tous les individus appartenant à l'ensemble *Archive* qui dominant la solution j .

Après cela, une phase de sélection d'individus qui vont entreprendre des opérations génétiques (croisement et mutation) est réalisée. Notons que dans cet algorithme, chaque individu présent dans l'*Archive* a de plus grande chance d'être sélectionné que les autres. Finalement, la population est remplacée par leurs descendants (résultat d'opérateurs génétiques).

Les principales différences entre les deux algorithmes SPEA et sa version améliorée SPEA-II sont comme suit :

- Dans SPEA-II, les individus présents dans l'*Archive*, ainsi que dans la population sont pris en compte lors de la phase d'affectation de valeurs de *fitness* (section 3-1 [170]),

- Contrairement à SPEA, dans SPEA-II la taille de l'*Archive* est fixée dès le début de l'algorithme, ce qui minimise les problèmes d'allocation de la mémoire,
- La technique de clustering utilisée dans SPEA quand la taille de l'ensemble des solutions non-dominées excède la taille de l'*Archive*, a été remplacée par une procédure de troncature qui permet de garder les points frontaliers dans l'*Archive*.

3.3.1.3 PESA-II

Cet algorithme a été proposé par Corne et al [166] en 2000. Comme ces prédécesseurs (PAES et SPEA) cet algorithme se base sur l'utilisation de deux ensemble d'individus : une population interne (P_I) et une externe (P_E). La population externe joue le même rôle que celui de l'*Archive* dans SPEA. Les similitudes les plus communes entre les trois algorithmes (PAES, SPEA et PESA) sont comme suit :

- Utilisation de deux ensembles de population : une interne qui regroupe les individus issus des re-combinaisons croisement et mutation, et une externe qui garde les solutions non-dominées obtenues jusque-là.
- Le processus de remplacement des individus dans l'ensemble externe est réalisé de la même manière dans les trois algorithmes. Ce processus est basé sur une mesure de surpeuplement.

La plus grande différence entre ces algorithmes réside dans l'affectation des valeurs de fitness et dans la mise à jour de l'*Archive*. Contrairement à SPEA qui utilise une valeur de puissance pour sélectionner (ou remplacer des individus dans l'*Archive*), PESA utilise un facteur de compression (*Squeeze Factor*) qui détermine si un individu va être sélectionné (enlevé de la population externe) ou pas.

Il existe aussi une version révisée de l'algorithme PESA appelé PESA-II. Cet algorithme est identique à l'ancien, à l'exception du fait que la sélection dans ce cas est basée sur la région. Les particularités de cette version de l'algorithme sont comme suit :

- Une sélection basée sur la région est utilisée plutôt qu'une méthode de sélection d'individus utilisée dans l'ancienne version de l'algorithme. Cette méthode sélectionne une hyperbox au lieu d'un individu. Après cela, les individus sont sélectionnés aléatoirement de l'hyperbox.
- Un facteur de compression (*Squeeze factor*) comme dans PESA est utilisé lors de la phase d'affectation de valeurs de fitness.

3.3.1.4 NSGA-II

La méthode NSGA-II initialement proposée par Deb et al [164] appartient à la famille des Algorithmes génétiques multi-objectifs basés sur la notion de dominance de Paréto. Cet

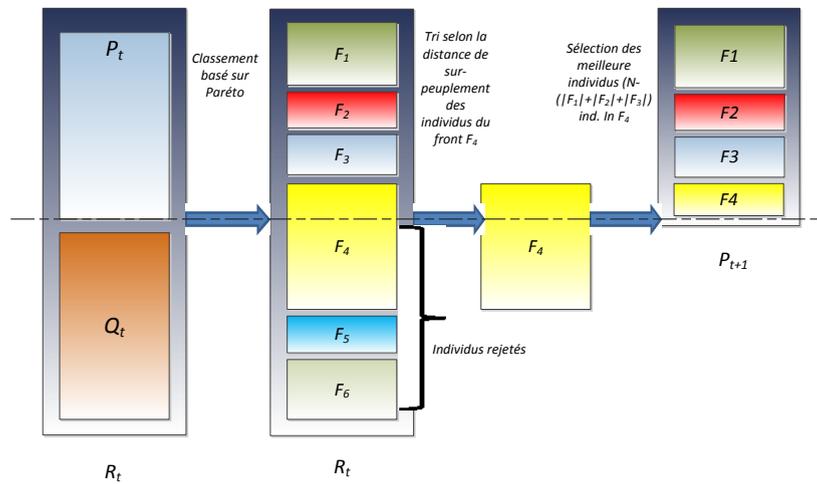


FIGURE 3.7: Architecture de l'algorithme NSGA-II

l'algorithme utilise une approche élitiste, sans paramétrage, qui manipule la population (solutions) en utilisant un mécanisme explicite afin de préserver sa diversité. Au début, l'algorithme génère N individus (généralement d'une manière aléatoire) qui forment la population initiale P_0 . Cette population est ensuite évaluée et classée selon la notion de dominance de Pareto. Les individus non-dominés sont assignés au premier front (F_1) et ont un rang égal à 1 ($rg = 1$). Après ceci, d'autres front sont formés en ignorant les individus précédemment choisis. Cette opération est répétée d'une manière récursive jusqu'à ce qu'il ne reste plus d'individus dans la population. Le principe de l'algorithme est représenté dans la Figure 3.7, ainsi que dans le pseudo algorithme 1.

3.3.1.5 Differential Evolution

L'évolution différentielle (Differential Evolution en anglais) a été introduite par Ken Price et Rainer Storn dans [167, 168]. L'algorithme effectue une recherche directe stochastique et appartient à la famille des algorithmes d'optimisation globale. L'algorithme d'évolution différentielle implique un maintien d'une population (solutions) qui participe (à chaque itération de l'algorithme) à un ensemble d'opérations : croisement, mutation et sélection comme dans les algorithmes évolutionnaires. La phase de croisement crée

Algorithm 1 NSGA-II

```

t ← 0
P0 ← InitialPopulation
while t < MaxGen do
  Rt ← Pt ∪ Qt
  F ← Tri des solution non – domines(Rt)
  Creation d'une population vide(Pt+1)
  i ← 0
  while |Pt+1| + |Fi| ≤ N do
    Pt+1 ← Pt+1 ∪ Fi
    Distance de surpeuplement Crowding – Distance(Fi)
    i ← i + 1
  end while
  Crowding – Distance(Fi)
  Pt+1 ← Pt+1 ∪ Fi[1 : (N – |Pt+1|)]
  Qt+1 ← Generer une nouvelle population avec les operateurs genetique Pt+1
  t ← t + 1
end while

```

une nouvelle solution en se basant sur trois individus de la population choisis au hasard. Ceci crée une perturbation dans la population pour mieux exploiter l'espace de recherche.

Dans l'évolution différentielle, la phase de mutation est accomplie en utilisant l'équation suivante :

$$v_{i,j} = y_{i,g} + F * (x_{r_1,g} - x_{r_2,g}) \quad (3.2)$$

Où $v_{i,g}$ est appelé vecteur de mutation. Les différents indices du vecteur solution $x_{r_i,g}$, r_1 et r_2 sont générés aléatoirement.

La phase de croisement effectue une combinaison des vecteurs $v_{i,g}$ et $x_{i,g}$ afin de générer un troisième vecteur $u_{i,g}$. Cette phase est accomplie en se basant sur l'équation suivante :

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{If } (rand_j[0,1] \leq Cr) \\ x_{j,i,g} & \text{otherwise} \end{cases} \quad (3.3)$$

3.3.2 Adaptation des algorithmes au problème de déploiement du RCSF

Afin de résoudre le problème de déploiement du RCSF, nous avons procédé à l'adaptation des différentes méthodes (NSGA-II, GA, SPEA-II, PESA-II, et DE) qui ont été présentées précédemment. Mise à part l'évolution différentielle (DE), les autres algorithmes offrent une facilité d'implémentation et s'adaptent aisément aux problèmes binaires. Leur plus grand défaut reste la création d'opérateurs adaptés qui conduisent à l'efficacité des algorithmes. Dans un ordre chronologique nous pouvons exprimer ces opérateurs comme suit :

- Premièrement, il faut définir une représentation adéquate des solutions appelées "Chromosomes".
- Une fois le codage de la solution défini, nous procédons à la création de la population initiale. En général, cette partie est réalisée d'une manière aléatoire.
- La population initiale créée, des opérateurs de reproduction sont appliqués afin de simuler l'évolution naturelle.

Pour l'évaluation différentielle (DE) nous avons développé une variante qui prend en charge l'optimisation des variables binaires. Cette variante a été inspirée des travaux effectués dans [171]. L'algorithme reste le même, mais les opérateurs de mutation et de croisement ont été changés.

3.3.2.1 Codage du chromosome

Dans notre cas les variables de décision sont des variables binaires. Nous pouvons les représenter selon deux manières possibles : (1) Vecteur réel de taille variable contenant les positions des différents capteurs, leur nombre est exprimé par la taille du vecteur, (2) Vecteur de taille fixe résultant de la translation de la grille de déploiement, où chaque position notifie la présence ou pas d'un nœud. Nous avons opté pour la deuxième solution, car elle facilite la création des opérateurs de mutation et de croisement. Un exemple d'individu est représenté dans la Figure 3.8.



FIGURE 3.8: Exemple d'une solution (individu)

3.3.2.2 Création de la population initiale

En général, cette étape est réalisée lors du démarrage des algorithmes évolutionnaires. Elle permet de générer un grand nombre de solutions réalisables, tout en maintenant une diversité suffisante dans la population, permettant d'éviter le piège des optima-locaux. Le but principal de cette phase est de trouver un équilibre entre diversité et solutions de bonnes qualités.

Dans notre cas, nous avons opté pour une génération aléatoire de la population initiale. Ce choix a été motivé par le fait que cette manière de générer la population ne consomme

pas trop de temps lors de l'exécution sur des grandes instances. Générer aléatoirement les individus permet de créer une diversité de la population tout en ne gardant que des solutions réalisables. La création d'un individu est comme suit : un capteur est ajouté d'une manière aléatoire dans la grille jusqu'à ce que cette dernière soit totalement couverte. Un exemple de population initiale avec cinq (05) individus est représenté dans la Table 3.2 :

Ind_1	1000110110
Ind_2	1010100101
Ind_3	0010011100
Ind_4	0100011001
Ind_5	1010001111

TABLE 3.2: Exemple d'une population initiale de cinq (05) individus

3.3.2.3 Sélection des individus

La phase de sélection définit le nombre de participation d'un individu à la reproduction dans une génération. Les individus dotés d'une meilleure valeur de fitness sont sélectionnés plus fréquemment que les autres. Les individus sélectionnés sont appelés parents et sont choisis pour participer à la phase suivante appelée reproduction. Dans notre cas, la sélection d'individus qui participent au croisement et à la mutation est réalisée grâce à la méthode "Tournoi". Deux parents sont sélectionnés au hasard dans la population et participent à un tournoi. Le parent doté de la meilleure valeur de fitness est sélectionné. Si les deux parents sélectionnés ont la même valeur de fitness, d'autres métriques sont rajoutées selon l'algorithme utilisé :

- NSGA-II : l'individu doté de la valeur minimale dite de *Crowding distance* est sélectionné
- SPEA-II ,PESA-II : on regarde le nombre d'individus dominés par la solution, le parent qui domine le plus d'individus est sélectionné.
- GA, DE : une variable aléatoire est générée. On procède après à la comparaison des variables, le parent doté de la valeur supérieure est gardé.

3.3.2.4 Opérateur de croisement

En règle générale, le croisement consiste à appliquer un processus avec une certaine probabilité (P_c) aux parents sélectionnés auparavant. Le taux de croisement (probabilité de croisement) définit la proportion de parent dans la population qui va être utilisée par un opérateur de croisement. Au début de l'algorithme, le croisement joue un rôle d'exploration de l'espace de recherche, car les nouvelles solutions générées par le croisement se

trouvent dans des espaces non explorés auparavant [169]. Dans notre algorithme, nous avons utilisé deux types classiques de croisement : croisement en un point. Le croisement utilisé est représenté par la Figure 3.9

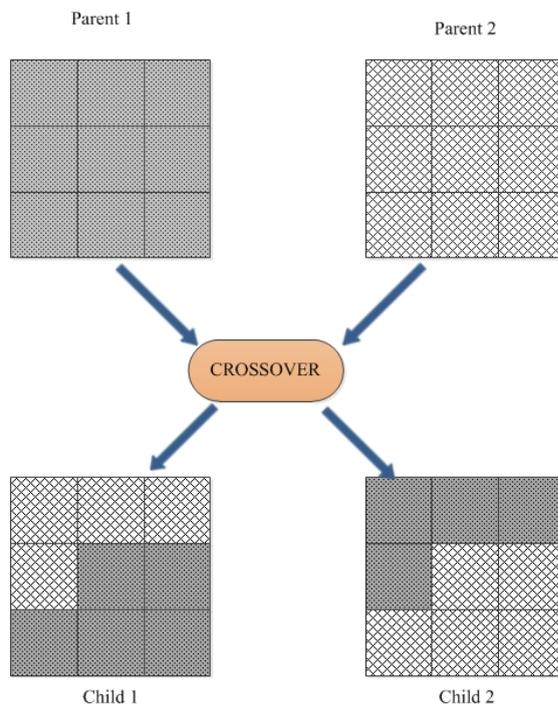


FIGURE 3.9: Exemple de croisement entre deux parents en un point

3.3.2.5 Opérateur de mutation

Une fois l'étape de croisement effectuée, l'opérateur de mutation est appliqué avec une faible probabilité (P_m) appelée taux de mutation. Cet opérateur introduit une perturbation dans la solution. Ceci conduit à maintenir la diversité de la population et explorer l'espace de recherche. Aussi, l'opérateur de mutation évite de converger vers des optima locaux. Nous avons opté pour l'utilisation d'une méthode binaire de mutation. La valeur d'un gène, choisi au hasard, est changé (0 s'il était 1, et 1 sinon). Ce type de mutation reste, dans le meilleur de notre connaissance, le plus adapté aux variables binaires. Un exemple de mutation est représenté à la Figure 3.10

3.3.2.6 Opérateur de réparation

L'opérateur sera utilisé lors de chaque étape de création des nouvelles solutions dans les méthodes évolutionnaires utilisées, à savoir : création de la population initiale, croisement, et mutation. Il garantit la faisabilité, par rapport au problème, des solutions

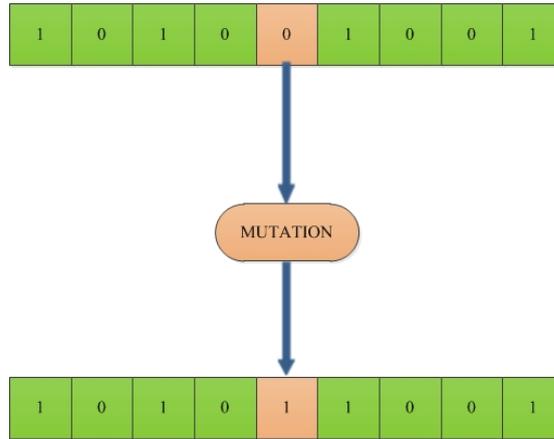


FIGURE 3.10: Exemple de mutation

trouvée jusque-là. l'opérateur que nous proposons se base sur les données relevées lors de la campagne de mesures. La première étape est de traduire ces données en des matrices, puis des vecteurs pouvant servir de base comparative entre la solution trouvée et la contrainte de faisabilité (algorithme 2). Ainsi, lors de chaque création de nouvelle solution, une comparaison en forme de multiplication élément par élément entre la solution et le vecteur de contrainte de faisabilité est réalisée selon l'algorithme 2. Ceci appliquera un masque sur la solution pour éliminer tous les emplacements impossibles (ex. : sur une fenêtre, porte, chauffage, ...) des capteurs ou routeurs. Le masque est conçu à partir du plan du bâtiment.

Algorithm 2 Opérateur de réparation

```

Sol ← Chargement de la solution existante
Vect ← Chargement du vecteur de contraintes
i ← 0
while i < Taille(Sol) do
  SolRepare(i) ← Sol(i) · Vect(i)
  i ← i + 1
end while
Return(SolRepare)

```

3.3.3 Métriques de comparaison

Afin d'avoir une étude comparative entre les différentes méthodes proposées, nous les avons développées sous MATLAB. La fonction objective a été modélisée par une somme pondérée des différents objectifs cités. Les algorithmes ont été exécutés sur la même

machine, afin que leurs performances soient équitablement évaluées.

$$\begin{aligned} Fitness(Sol_i) &= \omega_1 \cdot Cost_A(Sol_i) + \omega_2 \cdot CnCA(Sol_i) \\ &+ \omega_3 \cdot Cov_A(Sol_i) + \omega_4 \cdot Ov - Cov_A(Sol - i) \end{aligned} \quad (3.4)$$

Où :

$$\sum_i (\omega_i) = 1 \quad (3.5)$$

Les performances de la méthode de déploiement du RCSF sont évaluées en comparant les valeurs de fitness, ainsi que les temps d'exécution des différentes approches évolutionnaires. Les algorithmes ont été implémentés sur MATLAB, et sont exécutés sur la même machine qui est configurée comme suit : processeur Intel Core i7-3540M de 3.0 GHz, une RAM de 6 GB. Ceci nous permettra d'avoir une comparaison équitable pour tous les algorithmes.

Afin de calculer les différentes pondérations des fonctions objectives, nous avons développé un questionnaire dont l'interface est représentée sur la Figure 3.11. Ce questionnaire nous permettra d'établir un ordre de priorité des différents objectifs fixés et nous donnera les valeurs des différentes pondérations de chaque objectif. Le questionnaire a été développé

FIGURE 3.11: Interface graphique du questionnaire

sous MATLAB et déployé auprès de quelques personnes considérées comme des futurs utilisateurs de notre outil pour la phase test et validation. Comme on peut le voir sur l'interface (Figure 3.11), le questionnaire est divisé en deux phases. La première phase

nous permet d'établir un ordre de priorité des différents objectifs. La seconde, permet de déterminer les pondérations affectées à chaque objectif. Ces pondérations sont calculées selon l'équation 3.6. Cette équation nous permet de s'assurer que la contrainte $\sum_i \omega_i = 1$.

$$\begin{aligned}\omega_1 &\in [1/N_{Obj}, 1] \\ \omega_{i+1} &\in [(1 - \omega_i)/(N_{Obj} - i), \omega_i]\end{aligned}\tag{3.6}$$

3.4 Résultats et discussion

Afin de choisir un algorithme d'optimisation convenant au déploiement du réseau de capteurs sans-fil, nous avons procédé à une étude comparative entre les différentes méta-heuristiques définis auparavant à savoir : GA, NSGA-II, SPEA-II, PESA-II, et DE. Pour ce faire, nous avons identifié quatre objectifs, nécessaires au déploiement, qui constituent une base d'évaluation pour les différents algorithmes. Ces objectifs sont comme suit : (1) Coût, (2) Couverture, (3) Sur-couverture, et (4) Connectivité.

Le tableau 3.3 regroupe les principaux paramètres utilisés dans les différents algorithmes. Pour la couverture des évènements, ainsi que la connectivité nous avons utilisé les modèles présentés dans la section 2.3 (chapitre 2). Pour ce qui est du coût de déploiement, il est calculé selon l'équation 3.7. L'espace de déploiement où ont été testés les différents algorithmes, ainsi que les métriques utilisées seront présentées dans la section suivante (résultats et discussion),.

$$Cout = C_s * ||S|| + C_{Sk} * ||SK||\tag{3.7}$$

Où C_s, C_{sk} : représentent respectivement le coût d'achat et d'installation d'un nœud capteur ou routeur (sink). Pour des raisons de simplification, ces coûts sont considérés comme constants pour chaque type de nœud.

$||S||, ||SK||$: est la cardinalité des ensembles S et SK respectivement

3.4.1 Modélisation Mathématique

Soit A l'ensemble des positions possibles résultant de la discrétisation de la zone de déploiement en plusieurs cellules. Comme on l'a évoqué dans le chapitre précédent, chaque position $(i, j) \in A$ représente un placement possible d'un nœud du réseau (capteurs ou routeur). Une variable binaire $D_{i,j}$ associée à chaque position (i, j) de A , est mise à jour à chaque fois qu'un nœud est installé dans cette position. Pour des raisons de simplification nous posons X_i un vecteur binaire résultant de la translation de la matrice

Symbols	Meaning
$PL(d)$	Perte de signal à une distance d
$RSSI$	Received Signal Strength Indicator (indicateur de puissance du signal reçu)
L	Longueur du bâtiment
W	Largeur du bâtiment
d_0	distance de référence ($1m$)
L_{jk}	Facteur d'atténuation du mur i de type k
n	Facteur de perte de puissance
S	Ensemble des nœuds capteurs
SK	Ensemble des nœuds routeurs
R_s	Rayon de captage
P_{Tsh}	Seuil minimal pour pouvoir communiquer
$d(s, p)$	Distance entre le capteur s et la cellule $p_{i,j}$
Cov_A	Taux de couverture des évènements
CnC_A	Taux de connectivité
$Cost_A$	Coût de déploiement
$P_{TX/RX}$	Puissance de Transmission (TX) et réception (RX)
$P_{i,j}$	Cellule (i,j)

TABLE 3.3: Symboles utilisés

de déploiement par l'algorithme proposé dans la section 3.3.2.6 . Sk, S représentent respectivement l'ensemble des routeurs, et capteurs installés. La formulation des objectifs pour le placement de nœuds capteurs est comme suit :

$$\begin{aligned}
Minimize O_1 &= \sum_i (X_i) \\
Maximize O_2 &= Cov_A(X_i) \\
Maximize O_3 &= CnC_A(X_i) \\
Minimize O_4 &= Ov - Cov_A(X_i)
\end{aligned} \tag{3.8}$$

Sous contraintes :

$$\begin{aligned}
\sum_{i \in A} (a_{i,j} \cdot S_j) &\geq 1, \forall j \in \{1, \dots, |S|\} & (a) \\
\sum_{i \in S} (b_{i,j} \cdot Sk_j) &\geq 1, \forall j \in \{1, \dots, |Sk|\} & (b) \\
S \in A \text{ and } SK \in A && (c)
\end{aligned} \tag{3.9}$$

$$\begin{aligned}
Minimize O_1 &= \sum_i (X_i) \\
Maximize O_2 &= CnC_A(X_i)
\end{aligned} \tag{3.10}$$

$$\begin{aligned}
\text{Sous contraintes :} \\
SK \in A & \tag{d}
\end{aligned} \tag{3.11}$$

Durant la phase de déploiement des nœuds routeurs (Sinks), les deux seuls objectifs considérés sont : le coût et la connectivité. La durée de vie et la couverture sont ignorées dans cette phase, car un routeur est généralement alimenté par le circuit électrique du

bâtiment et ne remplit aucune tâche de surveillance de son environnement. Ces objectifs sont représentés par l'équation 3.10, et la contrainte par l'équation 3.11. Pour ce qui est du déploiement des nœuds capteurs, les objectifs 3.8 représentent respectivement : le coût, la couverture d'évènements, la connectivité, et finalement la sur-couverture. La contrainte 3.9.a exprime la nécessité de couvrir chaque position par au moins un capteur. Le coefficient binaire $a_{i,j}$ exprime la couverture de la position i par un ensemble de capteurs S (nœud capteur déployé dans A). Le coefficient $a_{i,j}$ est calculé grâce au modèle de couverture proposé dans la section 2.3. La contrainte 3.9.b, quant à elle, exprime la nécessité de connectivité de chaque capteur i appartenant à l'ensemble S par au moins un routeur Sk_j . Finalement, nous pouvons dire que le nombre de variables et de contraintes dépend fortement de la taille de la grille et de sa discrétisation.

Afin de valider les méthodes présentées précédemment, nous avons entrepris deux expérimentations sur des cas d'études réels. Les expérimentations menées sont définies comme suit :

- Expérimentation-I (Petite instance) : dans cette expérimentation, nous allons conduire un test des deux méthodes GA simple et NSGA-II afin d'étudier l'aspect multi-objectif du problème. L'expérimentation a été menée sur une partie du laboratoire LINEACT-CESI à Mont-Saint Aignan. L'espace de déploiement est constitué de deux bureaux séparés sur une surface de $100m^2$.
- Expérimentation-II (Grande instance) : Afin d'étudier la validité du modèle sur des bâtiments de grande taille, nous avons mené une expérimentation sur la totalité du bâtiment recherche du CESI Mont-Saint-Aignan. Ce dernier est constitué de plusieurs salles et bureaux qui occupent une surface de $850m^2$ sur deux étages.

3.4.2 Expérimentation I (Petite instance)

Dans un premier temps, nous allons tester les méthodes sur un espace restreint de $100m^2$. Ce dernier se compose de deux bureaux séparés comme le montre la Figure 3.12. Ce cas d'étude nous permettra de définir le paramétrage idéal des algorithmes étudiés à savoir : taille de la population, probabilités de croisement et mutation, et le critère d'arrêt.

3.4.2.1 Cas d'étude

Pour procéder à l'évaluation des différents algorithmes calculant le taux de connectivité, nous proposons de les tester dans un espace restreint (Figure 3.12) qui a été défini dans le chapitre précédent pour l'étude de la puissance de transmission (2.2.2). Cet espace représente $100m^2$ et se compose de deux bureaux séparés. Une campagne de mesures a été effectuée afin de déterminer les valeurs d'atténuations nécessaires pour le modèle de

connectivité. Le tableau 3.4 résume ces différentes valeurs d'atténuation, ainsi que les propriétés des capteurs/routeurs déployés.

Paramètre	Valeur
Longueur du bâtiment	10m
Largeur du bâtiment	10m
Puissance de transmission (routeur)	+2dBm
Puissance de réception (routeur)	-90dBm
Puissance de transmission (capteur)	+1dBm
Puissance de réception (capteur)	-90dBm
Atténuation des murs	3dBm
Atténuation de l'environnement (n)	2.8dBm

TABLE 3.4: Valeurs des paramètres utilisés



FIGURE 3.12: Vue 3D de l'espace de déploiement

3.4.2.2 Paramétrages des opérateurs

Afin d'identifier les bons paramètres à utiliser lors des phases de reproduction, nous avons étudié les probabilités de mutation et de croisement en fixant la taille de la population à 80 individus. Pour chaque couple de probabilités (P_c (croisement) et P_m (mutation)) l'algorithme génétique est exécuté plusieurs fois (minimum 3). La probabilité de croisement est variée de 0 à 0.9 avec un pas égal à 0.1, et celle de mutation varie entre 0 et 0.09 avec un pas de 0.01. Les résultats obtenus sont représentés dans les Table 3.5, Table 3.6, ainsi que les Figures 3.14 3.13.

Les résultats montrent que le meilleur couple de probabilités (mutation et croisement)

$\frac{P_m}{P_c}$	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0	140	84	73	85	77	93	131	120	101	106
0.1	75	72	63	64	72	83	74	85	95	97
0.2	77	54	55	57	68	70	70	75	86	87
0.3	63	51	49	51	51	59	66	63	68	75
0.4	48	46	45	50	52	51	59	62	61	62
0.5	43	41	43	45	47	48	50	53	55	55
0.6	40	42	42	43	43	45	47	49	51	51
0.7	42	39	39	40	42	42	42	45	48	48
0.8	40	37	37	37	37	40	40	42	42	43
0.9	37	37	37	37	37	37	37	37	39	39

TABLE 3.5: Impact de la variation des probabilités de mutation et de croisement sur le temps d'exécution (secondes)

$\frac{P_m}{P_c}$	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0	+13	-42	-52	-69	-69	-72	-64	-57	-52	-52
0.1	-52	-82	-82	-82	-81	-77	-75	-76	-65	-64
0.2	-47	-84.9	-84.9	-83	-84.9	-84.9	-81.4	-80.4	-79.85	-77.3
0.3	-64.7	-83	-83.95	-84.9	-83	-81.4	-83.9	-83.9	-83	-81.4
0.4	-74.8	-83.9	-84.9	-84.9	-84.9	-83.9	-84.9	-84.9	-84.9	-83.9
0.5	-77	-84.9	-84.9	-84.9	-84.9	-84.9	-84.9	-84.9	-84.9	-84.9
0.6	-78	-83.9	-84.9	-84.9	-83.9	-83.9	-83.9	-82	-84.9	-84.9
0.7	-81	-84.9	-84.9	-84.9	-83.9	-84.9	-84.9	-84.9	-82.3	-83
0.8	-80	-84.9	-84.9	-84.9	-84.9	-82.3	-84.9	-84.9	-84.9	-83.9
0.9	-83.7	-84.9	-84.9	-84.9	-84.9	-83	-84.9	-84.9	-84.9	-84.9

TABLE 3.6: Impact de la variation des probabilités de mutation et de croisement sur la valeur de fitness

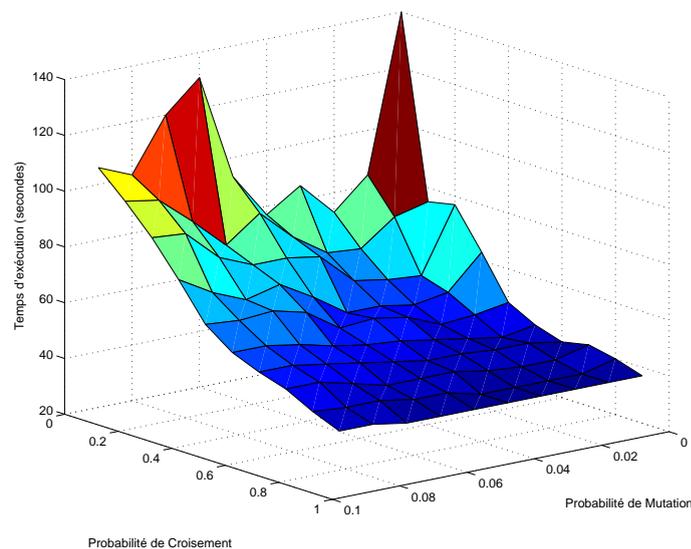


FIGURE 3.13: Variation de la probabilité de mutation

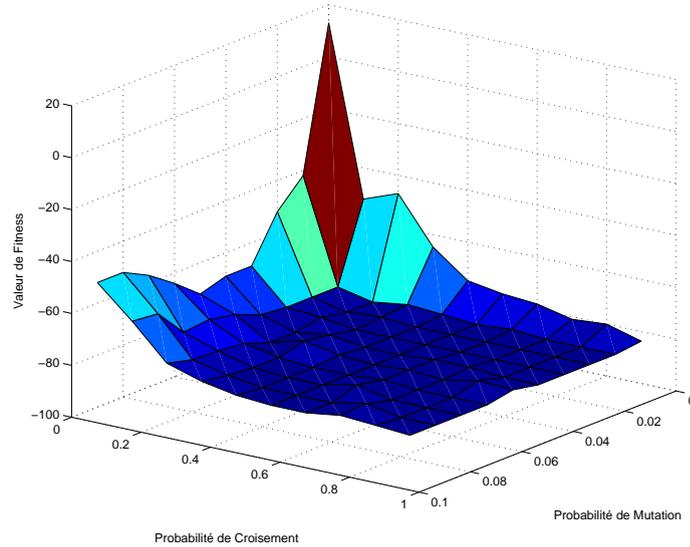


FIGURE 3.14: Variation de la probabilité de croisement

est dans l'intervalle $[0.8, 0.9]$ pour le croisement et $[0.01, 0.07]$ pour la mutation. Une fois le couple de probabilités identifié, nous avons procédé à l'identification de la taille de la population initiale optimale. Pour le faire, nous avons fixé la probabilité de croisement à 0.9 et celle de mutation à 0.07. Après avoir fixé les probabilités, nous avons étudié l'impact de la taille de population sur le temps d'exécution de l'algorithme génétique, ainsi que sur la valeur de fitness obtenue. La taille de la population a été modifiée entre 5 et 80 individus, et l'algorithme a été exécuté plusieurs fois (4 fois) pour chaque taille choisie. Les résultats obtenus sont représentés dans la Figure 3.15. Nous pouvons voir que l'algorithme converge quand on utilise 60, 70 ou 80 individus dans la population initiale. La différence entre ces trois tailles de population (60, 70, et 80) réside dans le temps d'exécution. Pour une population de 60 individus l'algorithme converge au bout de la 110ème génération. Pour les deux taille 70 et 80 individus, l'algorithme n'a besoin que de 51 génération pour atteindre l'optimum global (fitness = -84.9). En se basant sur ces résultats, nous avons défini une taille de population initiale égale à 70 individus et un couple de probabilités (mutation et croisement) égale à 0.9, 0.7 pour les autres algorithmes.

3.4.2.3 Algorithme génétique vs. NSGA-II

L'algorithme commence par générer des solutions aléatoires qui ne sont pas nécessairement exploitables. Après cela, et grâce aux opérateurs de reproduction, l'algorithme explore l'espace d'états en recherchant une solution optimale. Nous avons fixé notre

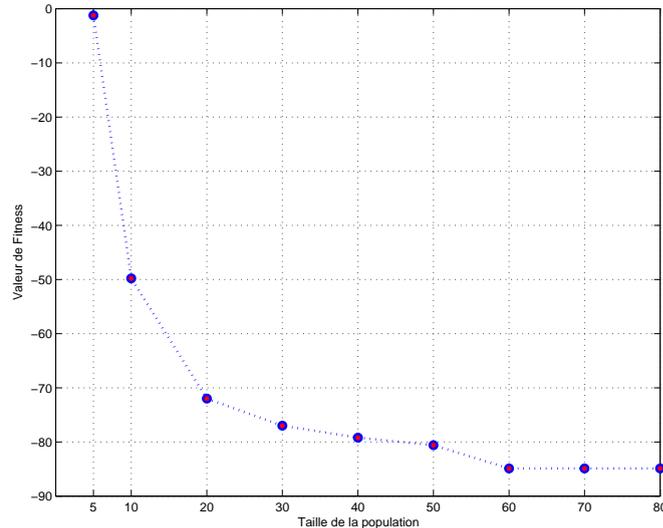


FIGURE 3.15: Impact de la variation de la taille de la population sur la valeur de fitness

critère d'arrêt pour les génération à 50, ceci veut dire qu'à chaque exécution de 50 générations l'algorithme inspecte la valeur de fitness, si cette dernière ne varie pas de plus de $1e-6$, l'algorithme génétique s'arrête, et la dernière valeur de fitness obtenue jusque-là est considérée comme la solution retenue. A partir de la Figure 3.16 nous pouvons voir que la valeur de fitness est minimisée à chaque génération jusqu'à ce que l'algorithme converge vers la 51ème génération. Ce graphique est obtenu grâce à la somme pondérée des différents objectifs présentée dans la section 3.3.3. La Figure 3.19 représente le front de Paréto obtenu après les 51 itérations de l'algorithme. La Figure 3.17 représente une comparaison entre deux solutions, la première est une solution tirée au hasard de la population initiale, et la deuxième représente la solution proposée par l'algorithme à la fin de son exécution. Nous pouvons voir qu'au début de l'exécution l'algorithme commence par déployer plusieurs nœuds (capteurs/routeurs). A ce moment, les objectifs de connectivité et de couverture sont atteints, sauf que l'objectif de coût et de sur-couverture reste inacceptable. A la fin d'exécution, la solution proposée comptabilise 4 nœuds routeurs et 10 nœuds capteurs. La valeur de chaque objectif est représentée dans la Table 3.7.

Objectif	Coût	Connectivité (%)	sur-couverture (%)	Couverture (%)
Valeur	10	96	3	91

TABLE 3.7: Valeurs des différents objectifs de la solution optimisée retenue

Dans l'objectif d'avoir une étude comparative entre l'algorithme génétique classique

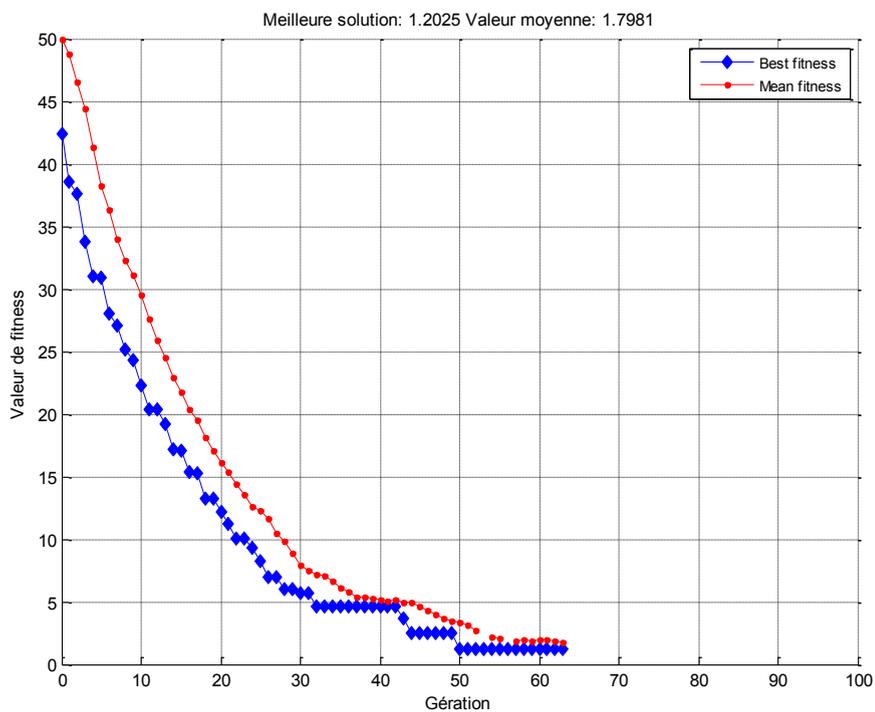


FIGURE 3.16: Évolution de la valeur de fitness)

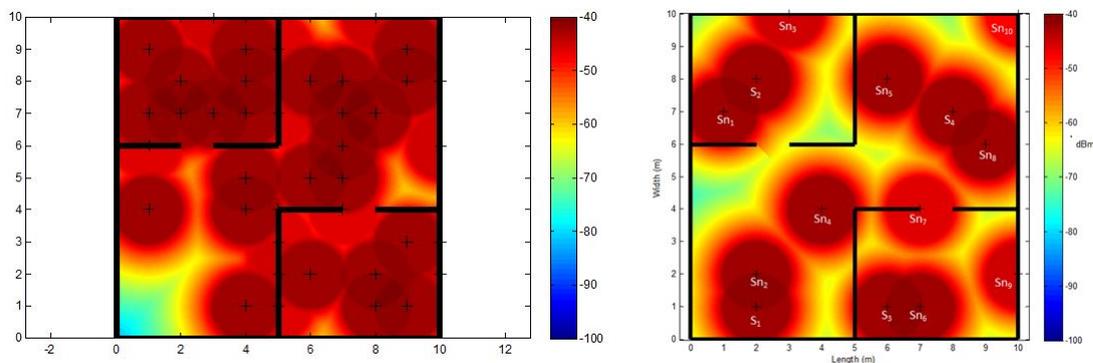


FIGURE 3.17: Solution issue de la première génération (gauche), et solution de la dernière génération (droite)

et l'algorithme NSGA-II, nous avons gardé les mêmes paramètres d'optimisation. Les probabilités de croisement et mutation ont été fixées à 0.9 et 0.03 respectivement. L'algorithme NSGA-II a été exécuté cinq (05) fois pour chaque configuration (taille de population variable) et les résultats sont représentés par la valeur moyenne issue des différentes exécutions. Le Tableau 3.8 et la Figure 3.18 montrent l'impact de la taille de la population sur le temps d'exécution de l'algorithme, nous pouvons déduire que ce dernier est proportionnel à la taille de la population. Aussi, plus petite est la population, plus l'algorithme converge vers des optimums locaux. Ceci est logique, car avec une population minimale l'algorithme ne peut pas garder une diversité dans la population, et ainsi ne peut plus explorer tout l'espace de recherche. L'objectif de cette première étude a été de choisir la meilleure taille de population initiale pour un minimum de temps d'exécution tout en gardant une convergence de l'algorithme. Finalement, notre choix s'est porté sur la même taille de population que l'algorithme génétique classique testé auparavant, c'est-à-dire 70 individus. Cette taille de population reste le meilleur rapport entre les deux paramètres : convergence et temps d'exécution.

A la fin de son exécution, l'algorithme NSGA-II fournit un panel de solutions groupées

$\frac{T_{pop}}{Simulation}$	Simulation-1	Simulation-2	simulation-3	simulation-4	simulation-5
10	31	23	31.4	26.6	26
20	27.3	31	32	26	28
30	52.3	36	42.3	35.5	41
40	42.2	44.2	46.5	48	47.2
50	44.2	45.6	50	47	42.7
60	54.4	49	46.7	49	53.8
70	64.8	61	54.6	52	57.6
80	75	75.6	75.5	72	72

TABLE 3.8: Impact de la taille de population sur le temps d'exécution

$\frac{Positions}{Sink}$	x	y
S1	2	1
S2	2	8
S3	6	1
S4	8	7

TABLE 3.9: Positions des nœuds routeurs (Sink)

dans le Front de Pareto. Ce front est représenté dans la Figure 3.19. Les solutions sont comme suit :

- **Phase de déploiement des nœuds Routeurs :** les solutions varient d'un nœud déployé (avec une couverture de 58%) jusqu'à quatre (04) nœuds routeurs déployés qui assurent une totale couverture de l'espace (Figure 3.21).

<i>Positions Capteurs</i>	x	y
Sn1	1	7
Sn2	2	2
Sn3	3	10
Sn4	4	4
Sn5	6	8
Sn6	7	1
Sn7	7	4
Sn8	9	6
Sn9	10	2
Sn10	10	10

TABLE 3.10: Positions des nœuds capteurs

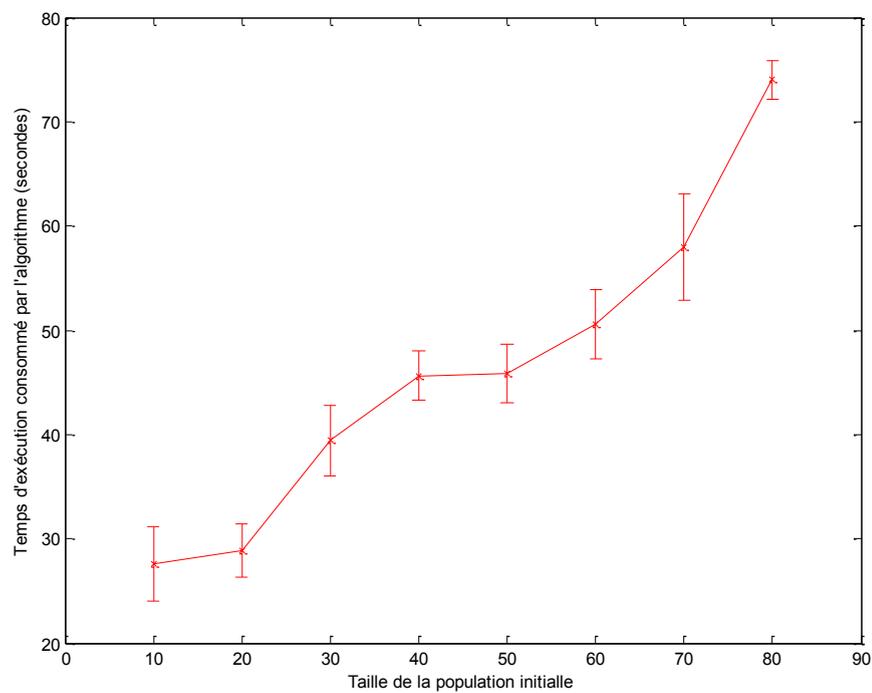


FIGURE 3.18: Taille de la population initiale vs. temps d'exécution

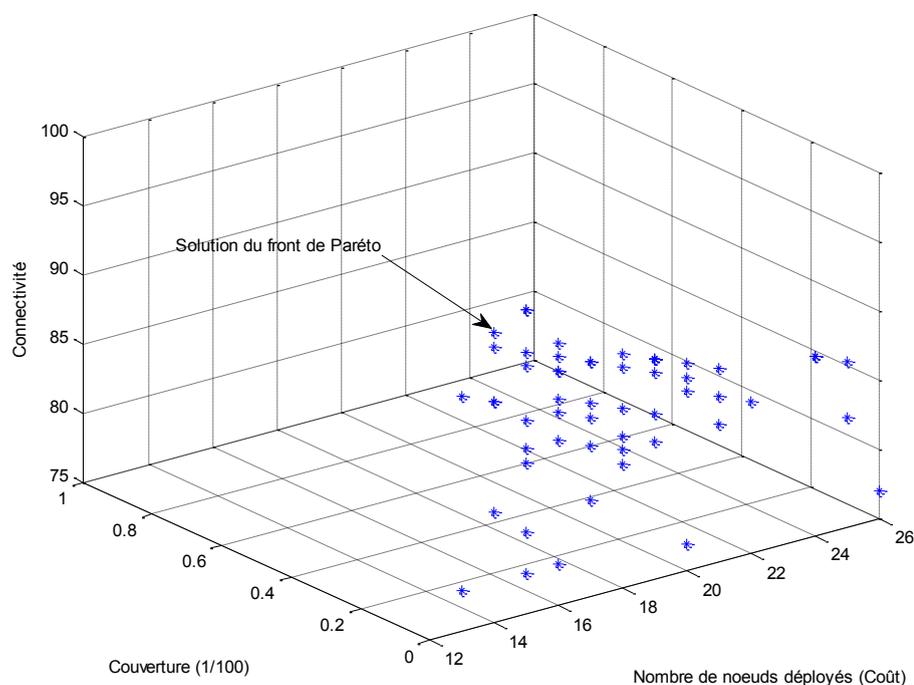


FIGURE 3.19: Front de Paréto de l'algorithme NSGA-II

- **Phase de déploiement des noeuds Capteurs :** En se basant sur les coordonnées des noeuds routeurs de la solution proposée précédemment (04 noeuds routeurs), nous exécutons une nouvelle fois l'algorithme NSGA-II afin d'obtenir la disposition des noeuds capteurs. Les solutions proposées par NSGA-II sont représentées par un Front de Pareto (Figure 3.22). Les topologies réseaux proposées varient entre quatre (04) noeuds déployés (avec une connectivité de 51%, et une couverture de 37%) jusqu'à 16 noeuds déployés qui assurent une couverture totale de l'espace avec une excellente connectivité (100%). Toutes les solutions proposées par l'algorithme NSGA-II sont exploitables, le choix est laissé à l'utilisateur et dépend de ses besoins. Si ce dernier a des contraintes financières et est flexible au niveau des objectifs couverture et connectivité, les solutions ayant moins de noeuds à déployer sont privilégiées (ex. : solution à 7 ou 8 noeuds capteurs assurant 80 à 90% de couverture). Si au contraire, l'utilisateur veut avoir une couverture totale du bâtiment et n'a pas de contrainte de coût, une topologie contenant 13 noeuds capteurs est possible. Cette dernière solution (13 noeuds capteurs et 4 noeuds routeurs) est représentée dans la Figure 3.20.

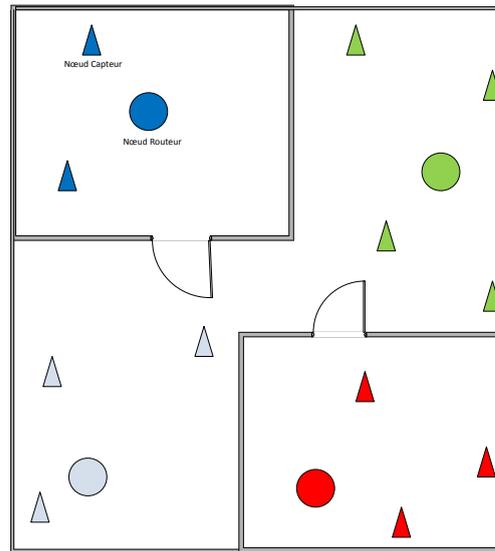


FIGURE 3.20: Capteurs et routeurs déployés selon la solution proposée par NSGA-II

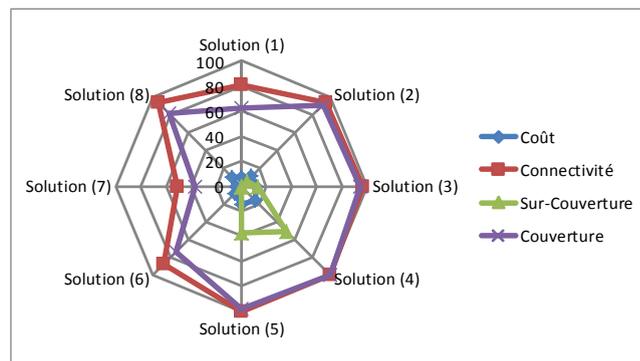


FIGURE 3.21: Front de Pareto obtenu pour le placement de noeuds routeurs

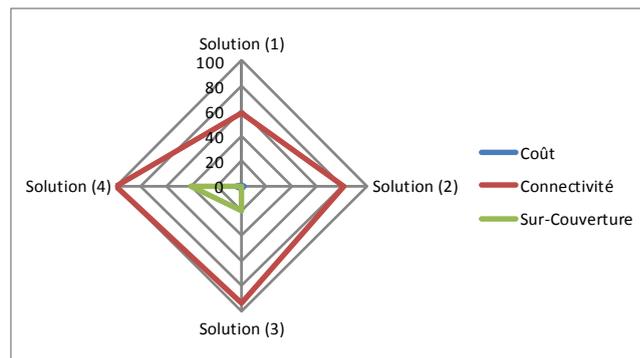


FIGURE 3.22: Front de Pareto obtenu pour le placement de noeuds capteurs

3.4.2.4 Validation et discussion

Afin de procéder à la validation de la topologie réseau proposée par les méthodes d'optimisation, nous avons développé un scénario de simulation sous NS-2.35 (simulateur réseau). Ce scénario consiste en une simulation du trafic réseau afin d'acquérir des métriques sur la qualité de service (QoS). Dans le Front de Pareto nous avons choisi une solution similaire à celle de l'algorithme génétique (10 nœuds capteurs et 4 nœuds routeurs). Nous avons utilisé le protocole Zigbee (IEEE 802.15.4) comme protocole de communication, et un modèle de propagation a été développé pour l'expérience. Le temps de simulation a été fixé à 600 secondes. Afin de simuler un trafic constant, nous avons utilisé un flux CBR (Constant Bit Rate) périodique avec un intervalle de 2.08 secondes entre les différents nœuds du réseau. Le débit de communication a été fixé à 256Kb/s et la taille du paquet à 128kb. Aussi, l'outil Visual Trace Analyser a été utilisé afin d'analyser les fichiers trace issus du logiciel NS-2.35 et contenant les résultats de simulation. Après l'exécution de la simulation, nous avons observé que la moyenne de délai entre l'envoi d'un paquet et sa réception est de 0.228 secondes avec une gigue (différence de délai de transmission de bout en bout entre des paquets choisis dans un même flux de paquets) de 0.0077 secondes, et ceci pour la totalité des nœuds du réseau. Le taux de délivrance des paquets est de 98.62% (pour un total de 368 640 paquets générés tout au long de la simulation, cf. Figure 3.23). Basé sur les travaux menés par [172], nous pouvons dire que les deux méthodes utilisées ont fourni des résultats valides qui répondent aux objectifs prédéfinis.

3.4.2.5 Comparaison avec d'autres Variantes d'algorithme évolutionnaires

En se basant sur les résultats obtenus auparavant (étude comparative entre NSGA-II et GA), nous avons procédé à l'implémentation de trois autres algorithmes : SPEA-II, PESA-II, et DE, afin d'étendre notre étude comparative et choisir la meilleure méthode à utiliser. Les différents algorithmes cités auparavant ont été implémentés sous MATLAB et exécutés sur la même machine pour avoir des résultats comparables. Les méthodes ont été évaluées sur trois critères principaux : temps d'exécution, temps cumulé, et nombre de générations nécessaire pour l'obtention d'un bon résultat. Les algorithmes ont été exécutés dix (10) fois chacun et la valeur de fitness pour chaque génération est calculée par la moyenne des résultats obtenus dans les dix exécutions. Les résultats obtenus sont représentés dans les Figures 3.24, 3.25, et 3.26. Nous pouvons voir sur le premier graphique (Figure 3.24), que tous les algorithmes implémentés finissent par avoir le même résultat final à la fin de leurs exécutions (résultat représenté dans la Figure 3.17). Aussi, ce graphique nous montre l'évolution de la valeur de fitness pour chaque algorithme

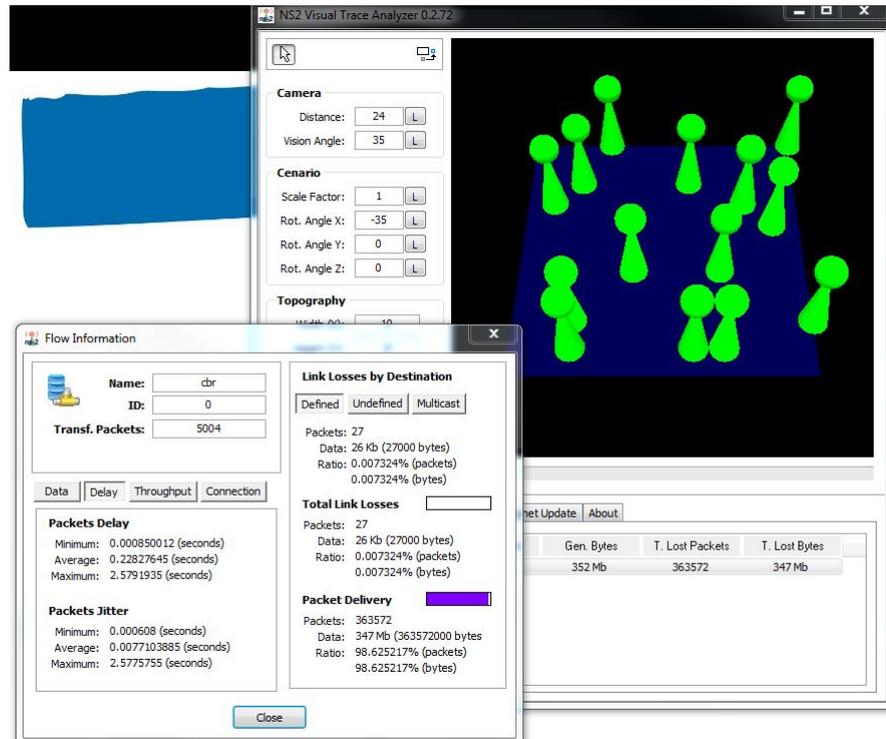


FIGURE 3.23: résultat de la simulation sur NS-2 (données traitées par Visual Trace Analyser)

durant son exécution (valeur de fitness dans chaque génération). Nous pouvons déduire que l'algorithme génétique est le plus performant en terme de convergence rapide, car il atteint la valeur de fitness optimale au bout de 19 générations. Pour le reste des algorithmes, la convergence varie entre 32^e et 67^e générations (PESA-II : 32, SPEA-II : 54, et DE : 67). Ceci est dû au nombre d'opérations exécutées par chaque algorithme et durant chaque génération. L'algorithme génétique ne nécessite pas d'opérations de calcul de distance ou de clustering comme dans le reste des algorithmes. Le dernier des algorithmes à converger est l'évolution différentielle (DE), le résultat est obtenu après 67 générations, nous observons aussi qu'à plusieurs périodes l'algorithme reste figé dans un optimum locale. Les notions de reproduction et de sélection utilisées dans DE devraient être améliorées pour permettre une meilleure convergence et une bonne exploration de l'espace de recherche.

Les deux autres figures (Figure 3.25, et Figure 3.26) présentent les performances temporelles de chaque algorithme. La première figure montre l'évolution de la valeur de fitness dans le temps. Nous pouvons voir sur le graphique que l'algorithme génétique et l'algorithme PESA-II restent les meilleurs en termes de performances. Aussi, nous pouvons voir que les algorithmes n'ont pas le même comportement, ils diffèrent dans le temps consommé dans chaque génération. Chaque algorithme commence par consommer beaucoup de temps pour traiter les individus et calculer leurs valeurs de fitness dans les

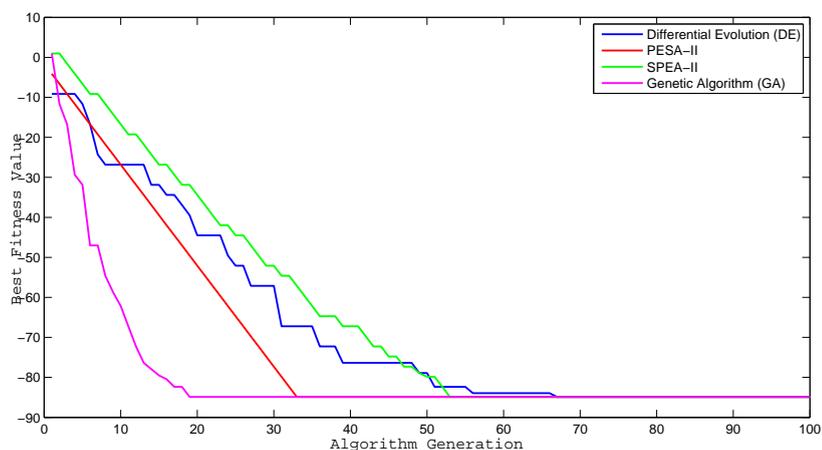


FIGURE 3.24: Évolution de la valeur de fitness vs. Génération des algorithmes

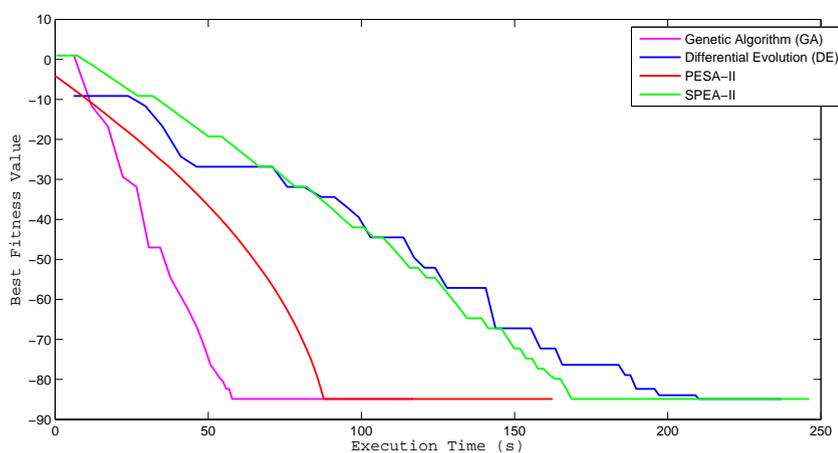


FIGURE 3.25: Évolution de la valeur de fitness vs. temps d'exécution

premières générations, mais arrivant à une certaine valeur le temps passé dans chaque génération diminue. Nous observons aussi une nette différence entre la comparaison des algorithmes en termes de générations d'exécution et en termes de temps d'exécution. Pour l'algorithme DE la convergence se fait après celle de SPEA-II, sauf que quand on regarde les performances des mêmes algorithmes en termes de temps d'exécution, DE a tendance à finir son exécution bien avant l'algorithme SPEA-II. Comme conclusion, nous pouvons dire que l'utilisation d'une méthode ou d'une autre est liée aux besoins de l'utilisateur. Pour ce qui est des algorithmes génétiques avec somme pondérée, l'utilisateur doit définir les différentes pondérations des objectifs avant le lancement de l'algorithme. Ce choix est crucial, car la solution proposée par la méthode dépend fortement de ces pondérations. Cet algorithme reste un très bon choix si l'utilisateur sait parfaitement quelle objectif il veut optimiser et avec quel pourcentage. Pour ce qui est de

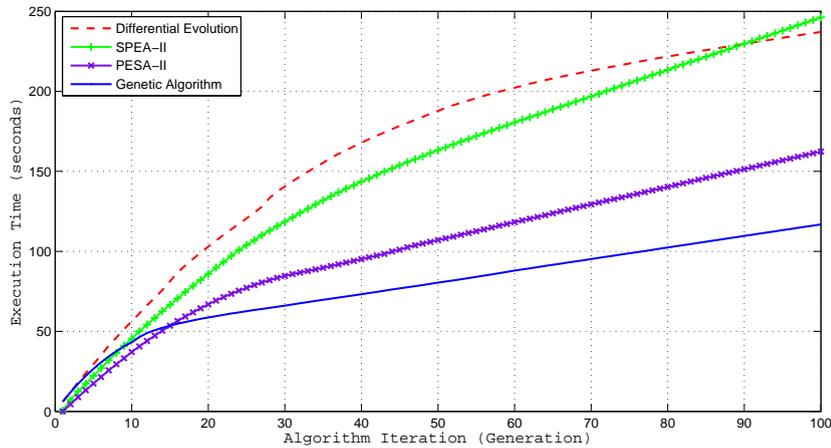


FIGURE 3.26: temps d'exécution cumulé dans chaque génération

la méthode NSGA-II, elle fournit des solutions variées et réalisables groupées dans le front de Pareto. L'utilisateur peut ainsi parcourir ces solutions et choisir celle qui lui va le mieux. La seule lacune de cette méthode est le temps d'exécution. En effet, la méthode NSGA-II consomme plus de temps d'exécution car elle procède à un très grand nombre d'opérations (classement, crowding distance, ...) qui ne sont pas réalisées par l'algorithme génétique (avec somme pondérée).

Finalement, le choix de la méthode d'optimisation utilisée dépend fortement de l'utilisateur final de l'outil de déploiement. Si l'utilisateur connaît a-priori les objectifs visés, ainsi que leur importance (pondérations), nous recommandons l'utilisation de l'algorithme génétique avec somme pondérée. Sinon, l'utilisation d'un algorithme fournissant un front de Pareto (ex. : NSGA-II) est fortement recommandé.

3.4.3 Expérimentation II

Afin d'étudier le comportement des méthodes d'optimisation dans des environnements plus grand, nous proposons de les tester sur un plus grand bâtiment. Le bâtiment choisi est une partie du laboratoire de recherche de l'école d'ingénieurs du CESI-Rouen. L'espace comptabilise une surface totale de $850m^2$ (25 m de longueur, et 34 m de largeur). Le bâtiment est composé de plusieurs bureaux, un espace ouvert dédié à la recherche, ainsi qu'un FabLab fixe regroupant des outils de modélisation et fabrication d'objets 3D (Figure). Une campagne de mesures et de relevés d'informations a été conduite afin de déduire les contraintes de placement (Figure 3.30 et Figure 3.31). Nous avons relevé au total :

- 43 prises électriques pouvant accueillir des noeuds Zplug (capteurs de consommation électriques et routeurs),

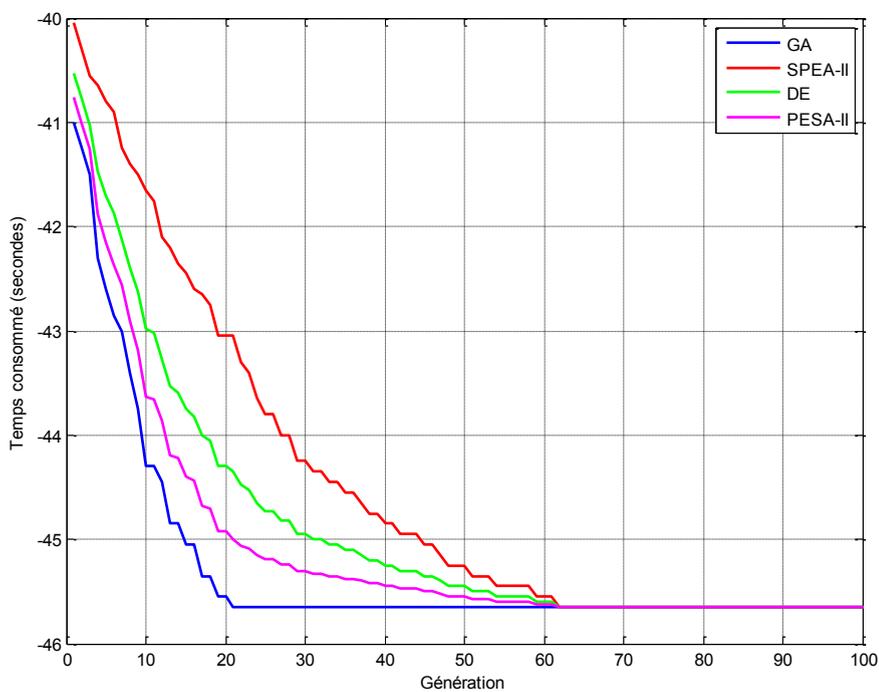


FIGURE 3.27: Valeur de Fitness VS. Génération (déploiement capteurs)

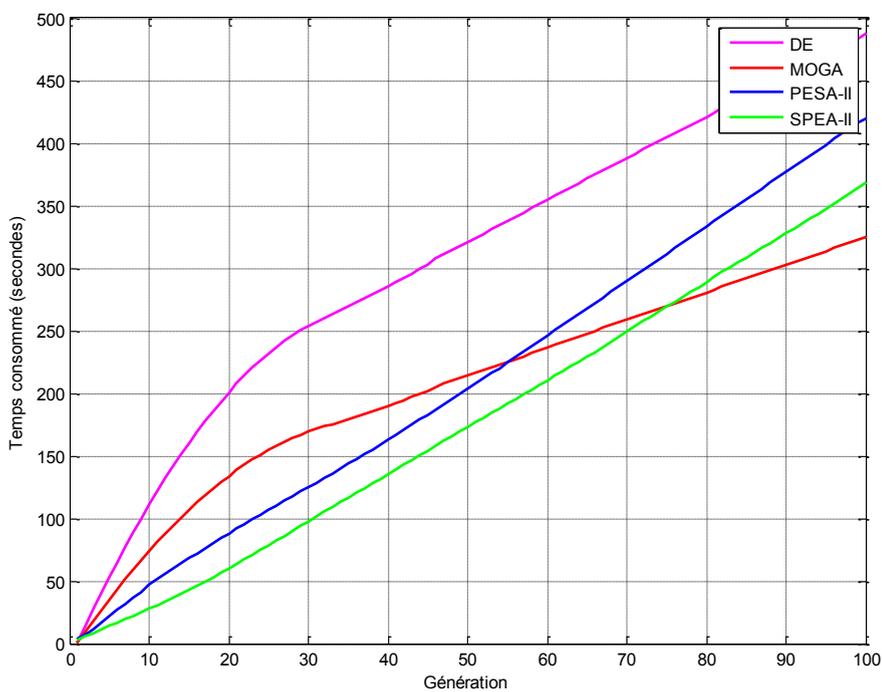


FIGURE 3.28: Temps consommé dans les génération de l'algorithme (déploiement capteurs)

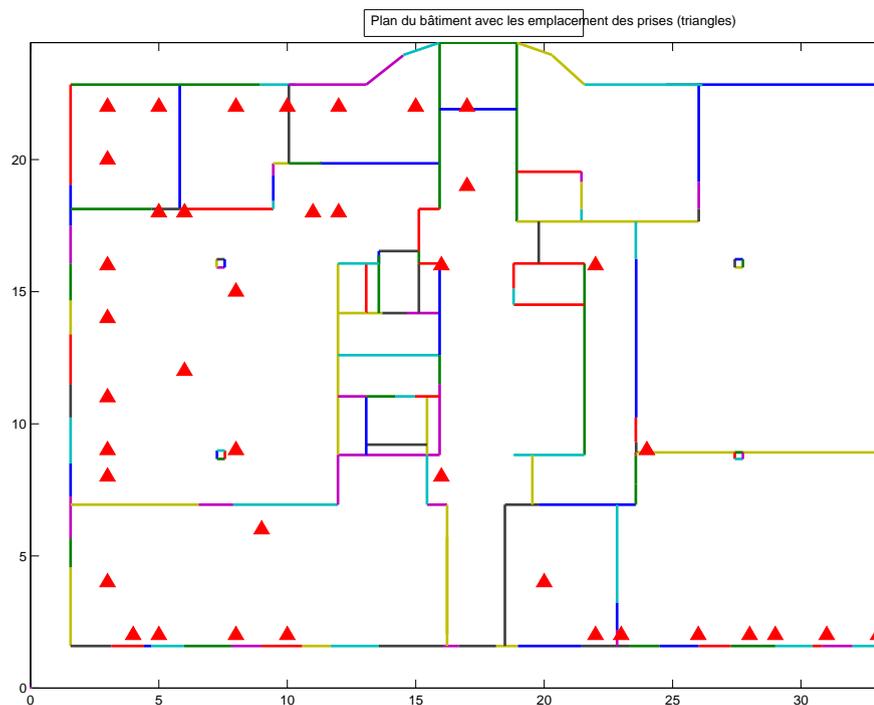


FIGURE 3.29: Plan du bâtiment avec les emplacements des prises

- Les différentes sources de chaleurs (chauffages) et d'air conditionné ont été recensées (27),.
- Les impossibilités de placement (portes, fenêtres, meubles, ...) de nœuds capteurs (Zlum, ZHum, ...) ont été relevées,

Le plan du bâtiment, ainsi que les différents paramètres relevés sont représentés dans la Figure 3.29.

La Figure 3.30 montre la valeur de l'atténuation radio des différents matériaux utilisés dans la construction de bâtiment. On peut voir sur la figure que la valeur d'atténuation varie selon le type et l'épaisseur du matériau utilisé. Pour une porte en bois cette valeur est d'environ $3dBm$ si la fréquence utilisée est égale à $900MHz$ et de $3.2dBm$ si la fréquence est égale à $2.4GHz$. Aussi, on peut voir que pour le même matériau utilisé la valeur d'atténuation ne reste pas la même et varie selon l'épaisseur. Par exemple, pour un mur construit en béton, on peut trouver différentes valeurs d'atténuations qui dépendent de son épaisseur (ex. : béton (203mm) a une atténuation de $12.8dBm$, béton (305mm) a une atténuation de $35dBm$). À travers ce graphique, nous pouvons déduire que la valeur d'atténuation est un paramètre qui dépend fortement de : (1) caractéristiques du matériau, et (2) son épaisseur. Vu que notre réseau utilise le protocole Zigbee avec une fréquence de $2.4GHz$ nous allons nous intéresser, pour la suite, à l'atténuation radio dans notre bâtiment à $2.4GHz$. Pour le faire, nous avons utilisé les capteurs développés

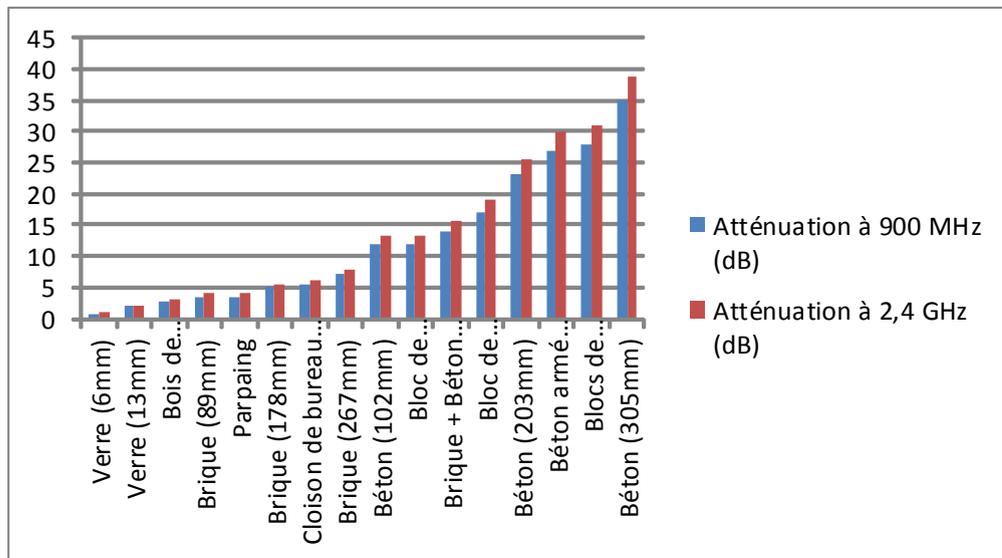


FIGURE 3.30: Valeurs d'atténuations radio des matériaux selon[14]

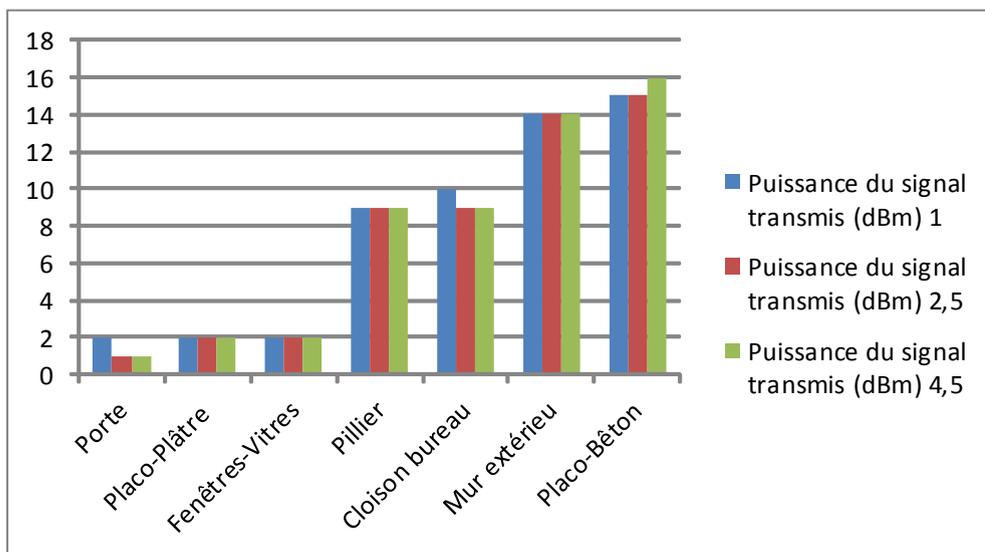
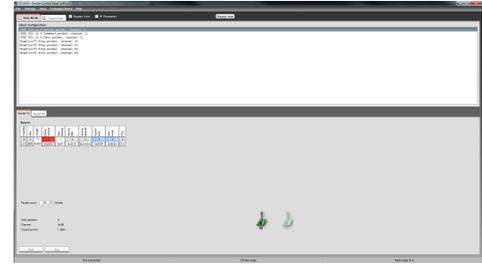


FIGURE 3.31: Valeurs d'atténuations radio des matériaux dans le laboratoire IRISE du CESI



(A) Interface principale



(B) Interface de gestion des équipements

FIGURE 3.32: Différentes vues de l'interface de l'outil SmartRF-Studio (TI)

dans notre laboratoire et qui ont été exposés dans le chapitre précédent. À l'aide de l'outil SmartRF-Studio(cf. Figure 3.32) nous avons mené des expériences afin de déduire les atténuations radio des différentes cloisons présentes dans le laboratoires. Pour chaque cloison (porte, fenêtre, mur, ...) nous avons procédé comme suit :

- Deux kits de développement Texas Instrument (TI) programmés (cf. chapitre 2) sont placés de chaque côté de la cloison,
- Un kit est programmé pour être un nœud émetteur et l'autre un nœud récepteur,
- Le kit récepteur est connecté à un ordinateur (équipé de l'outil SmartRF-Studio) afin de collecter la puissance du signal reçu,
- Pour déduire la valeur d'atténuation nous appliquons l'équation qui définit la propagation radio (MWF) exposée dans le chapitre 2.

Cette expérience est menée plusieurs fois (06 fois) pour chaque cloison et la valeur finale est calculée par la moyenne de toutes les valeurs relevées. À la fin, toutes les valeurs ont été regroupées dans un tableau, et sont représentées graphiquement sur la Figure 3.31. Ces valeurs ont été intégrées dans l'algorithme d'optimisation pour le calcul de la connectivité.

Une fois que tous les paramètres ont été relevés et intégrés à l'algorithme, nous lançons les méthodes d'optimisations. L'opération effectuée pour la petite instance (bâtiment de $100 m^2$) est renouvelée afin d'avoir une étude comparative entre les méthodes étudiées. Les résultats obtenus sont représentés dans les Figures 3.33, 3.34 et la Figure 3.35.

La Figure 3.33 montre l'évolution de la valeur de fitness (fonction objective) durant chaque génération des algorithmes étudiées à savoir : GA, DE, SPEA-II, et PESA-II. Nous pouvons voir que seuls les deux algorithmes PESA-II, et SPEA-II atteignent des valeurs significatives (optimum global), alors que les algorithmes GA, et DE restent dans l'optimum local et n'arrivent plus à améliorer la solution à partir d'une certaine génération. Ceci est du au comportement des algorithmes. Pour SPEA-II et PESA-II,

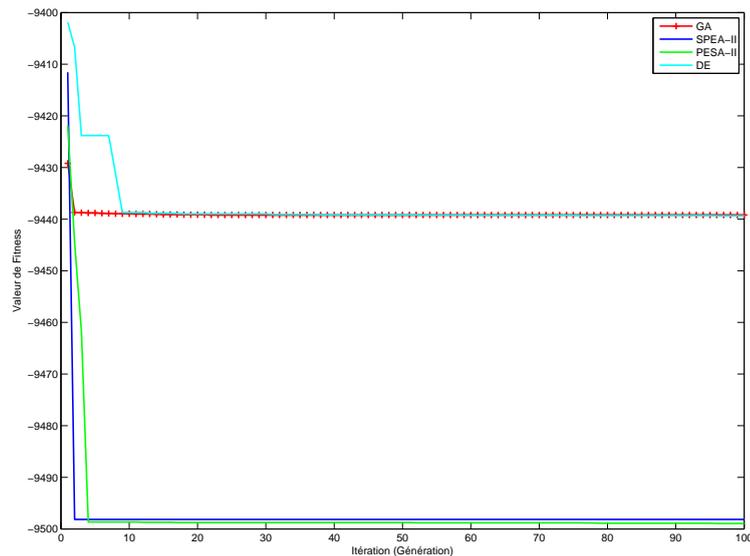


FIGURE 3.33: Évolution de la valeur de fitness vs. Génération des algorithmes (Grande instance)

le choix des solutions dans la phase de sélection est basé sur la dominance selon *Pareto* ou des techniques de clustering sont utilisées pour la réduction du nombre d'individus dans la base *Archive* (ex. : supprimer les doublons, ...). Alors que les algorithmes GA et DE, n'utilisent aucune notion de dominance et se base sur les valeurs de la fonction objective pour le choix des individus dans la phase de sélection. Ceci nous conduit à la conclusion suivante : *"Le choix des algorithmes à utiliser dépend fortement de la taille du problème. Si le problème est de petite taille (petite instance) l'utilisation d'un algorithme de type P-métaheuristique simple tels que GA ou DE est préféré, car la solution obtenue est de bonne qualité trouvée dans un temps minimal. Si la taille du problème devient très grande (grande instance) l'utilisation des algorithmes qui se basent sur la notion de dominance est fortement recommandée"*.

La Figure 3.34 représente le temps d'exécution consommé par chaque algorithme pour 100 générations. Nous pouvons voir que l'algorithme PESA-II est plus performant que les autres, non seulement PESA-II fournit la meilleure solution (surpasse SPEA-II), mais aussi consomme moins de temps que le reste des algorithmes utilisés. Le temps que consomme PESA-II est d'environ 3000secondes (à peu près comme celui consommé par GA), alors que DE exécute 100 générations en 3500secondes et SPEA-II près de 4250secondes. Aussi, nous pouvons voir que les deux algorithmes SPEA-II et PESA-II trouvent la meilleure solution dans les premières générations. Ceci vient confirmer la conclusion citée précédemment, dans ce genre de problème (espace de recherche très grand, taille de solution très grand (850 variables), ...) le choix d'algorithmes est crucial.

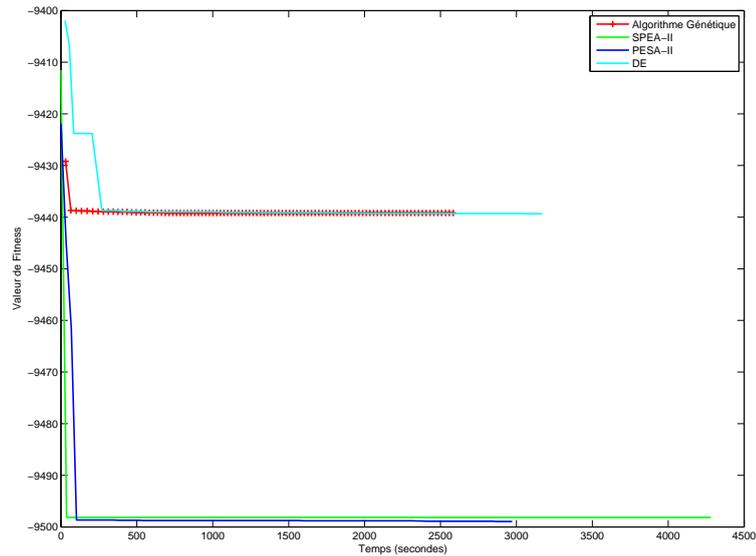


FIGURE 3.34: Évolution de la valeur de fitness (somme pondérée) dans le temps d'exécution (Grande instance)

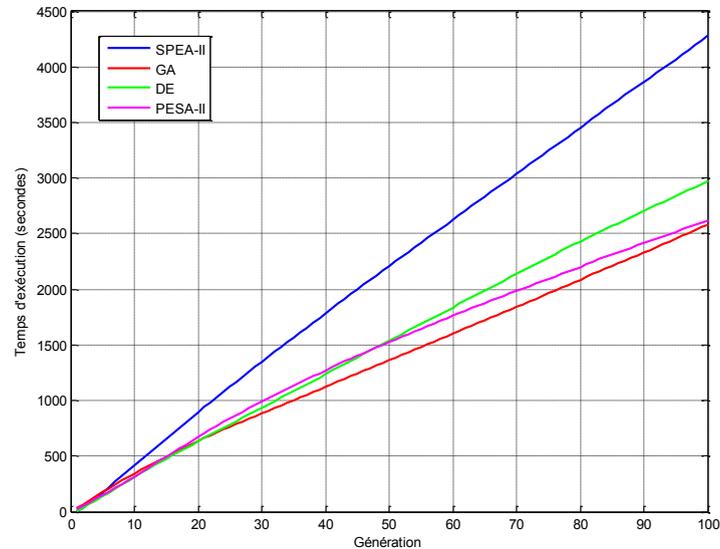


FIGURE 3.35: temps d'exécution cumulé dans chaque génération (Grande instance)

La dernière Figure 3.35 montre le temps consommé durant l'exécution de chaque génération de l'algorithme. Les deux extrêmes sont les algorithmes GA (plus rapide) et SPEA-II (le plus gourmand en temps). Entre les deux, on trouve l'algorithme PESA-II et l'algorithme DE qui comptabilise 3000 et 3200 secondes consommées pour trouver la solution du problème.

Finalement, nous pouvons dire que notre choix se portera sur l'algorithme GA pour le déploiement en espace restreint (moins de $100m^2$), et l'algorithme PESA-II pour des espaces plus grands. Ce choix a été la conclusion des travaux menés jusque là et des tests menés dans notre laboratoire. Aussi, nous remarquons que les algorithmes utilisés fournissent des solutions au bout d'une heure d'exécution (3000-3600 secondes d'exécution). Ceci reste très lent du point de vue utilisateur, et nécessite une amélioration

des algorithmes pour qu'ils puissent trouver une topologie réseau dans de plus brefs délais.

3.5 Conclusion

Dans ce chapitre nous avons présenté le problème d'optimisation de déploiement d'un RCSF, ainsi que les différentes méthodes pouvant le résoudre. Dans la première partie de ce chapitre nous avons commencé par définir un problème d'optimisation multi-objectif ainsi que toutes les méthodes existantes dans la littérature. Vu le grand nombre de méthodes de résolution proposées dans la littérature, et vue que notre problème est NP-Difficile, nous avons restreint notre étude au méta-heuristique. Nous avons sélectionné cinq d'entre elles qui sont : Algorithmes Génétiques, NSGA-II, Évolution Différentielle, SPEA-II, PESA-II. Ces cinq méthodes restent, au meilleure de notre connaissance, les meilleures et les plus performantes dans le domaine de l'optimisation multi-objectif.

Dans la deuxième partie du chapitre, nous avons présenté un état de l'art sur les méthodes d'optimisation utilisées dans le problème de déploiement des RCSF, en montrant la relation entre ces méthodes et les méthodes que nous avons sélectionnées. Après cela, nous avons expliqué en détail la méthodologie de déploiement suivie (déploiement des routeurs et déploiement des capteurs). Aussi, les fonctions objectives et les différentes contraintes considérées ont été présentés dans cette partie. Finalement, nous avons terminé cette partie par la présentation des différents paramètres utilisés par les méthodes d'optimisation à savoir : codage du chromosome, création de la population initiale, sélection, ...

Une fois les méthodes d'optimisation sélectionnées, et leurs paramètres définis, nous avons procédé à deux types d'expérimentations. La première se résumait dans le déploiement d'un RCSF dans un espace restreint ($100m^2$). Afin d'y parvenir, nous avons commencé par la collecte des données sur l'environnement de déploiement (valeurs d'atténuation, ...). Une fois ces paramètres connus, nous avons lancé les deux algorithmes GA, et NSGA-II afin d'étudier l'apport scientifique de la notion de dominance selon Pareto. Nous avons conclu cette expérience en expliquant que chaque algorithme à ses propres caractéristiques et dépend du choix de l'utilisateur. Après cela, nous avons procédé à une étude comparative entre tous les algorithmes sélectionnés. D'après les résultats retrouvés, nous avons conclu que l'algorithme génétique (GA) est l'algorithme le plus approprié pour ce type de problèmes (déploiement en espace restreint). Cet algorithme fournit de très bons résultats en un minimum de temps.

Afin de confirmer les hypothèses données par la première expérience, une deuxième expérience a été menée sur un espace plus grand ($850m^2$).

Chapitre 4

Méta-heuristique hybride pour le déploiement des RCSF

4.1 Introduction

Dans le chapitre précédent nous avons présenté une étude comparative entre différentes méthodes d'optimisation multi-objectifs (GA, DE, NSGA-II, SPEA-II, et PESA-II) utilisées pour la résolution du problème de déploiement des RCSF (NP-Difficile). Les résultats trouvés nous ont permis d'identifier les algorithmes les plus performants, ainsi que les cas d'études où ils peuvent être utilisés. Nous avons déduit que l'utilisation d'une *P-métaheuristique* simple (sans notion de dominance) comme l'algorithme génétique est préférable quand le problème est de petite taille (ex. : bâtiment de $100m^2$). Ce type d'algorithmes devient gourmand en temps de calcul et en mémoire, et a tendance à ne pas converger vers l'optimum global dès que la taille du problème devient trop grande. Pour cela, nous avons testé des *P-métaheuristique* intégrant une notion de dominance (Paréto), les résultats trouvés montrent que ce type d'algorithme est mieux adapté aux problèmes *NP-Difficile* de grande taille (ex. : bâtiment de $850m^2$).

Par contre, nous avons remarqué une lenteur d'exécution lors de la phase test sur un grand bâtiment (espace de déploiement très grand). Les algorithmes utilisés fournissent une solution après $4000secondes$ d'exécution. Ceci reste très grand vis-à-vis de l'application qu'on se propose de développer. Il faut que l'utilisateur (un technicien d'installation de réseau) puisse calculer rapidement les positions de ses nœuds en très peu de temps et effectuer rapidement plusieurs essais. Il n'est pas envisageable qu'un utilisateur attende plus d'une heure pour avoir apparaitre sa solution. De plus l'utilisation d'un super ordinateur n'est pas envisagé dans ce genre d'utilisation. Un outil avec un temps de calcul très important ne sera pas intéressant pour de tels utilisateurs. Pour palier ce

problème, nous proposons, dans ce chapitre, une amélioration des algorithmes testés auparavant en réduisant le temps de calcul et en gardant la même qualité de solution. Cette amélioration se résume en une limitation des fonctions d'évaluations qui prennent beaucoup de temps pour évaluer chaque individu. Les algorithmes proposés dans ce chapitre se basent sur la notion de data-mining et de clustering pour définir des individus porteurs d'informations. Une fois ces individus identifiés, une étape d'évaluation est appliquée à ces derniers grâce à (aux) la fonction(s) objective(s) (évaluation directe). La valeur de fitness des individus restants (reste de la population) se voit attribuer une valeur approximative qui se base sur la valeur de fitness de l'individu porteur d'information et la notion des RN (Réseau de Neurones). Ces algorithmes sont appelés algorithmes hybride ou métaheuristiques hybrides. D'un autre côté, nous proposons dans ce chapitre une étude de la fiabilité du réseau en utilisant les coefficients de connexité. Ce dernier nous permet de calculer le nombre de liens existants et d'en choisir un à chaque fois qu'un nœud tombe en panne.

4.2 Analyse des temps consommés

Afin de déterminer l'opération qui consomme la plus grande proportion de temps, nous effectuons un prélèvement des temps consommés dans chaque itération par chaque opération de l'algorithme SPEA-II. Le choix de l'algorithme à surveiller est logique, car cet algorithme semble le plus performant, en termes de solution et de temps d'exécution, pour des espaces très grands (d'après les résultats du chapitre précédent). La mesure des durées est effectuée sur les principales opérations des algorithmes évolutionnaires cités auparavant, à savoir : classement des solutions (dominance selon Paréto), croisement, mutation et enfin l'évaluation. L'analyse des temps consommés se passe comme suit :

- Avant chaque itération le temps est initialisé ($temps_d = \text{heureactuelle}$),
- À la fin de l'exécution de l'opération (classement, croisement, mutation et évaluation) le temps (heure du processeur) est prélevé ($temps_f$),
- La durée consommée par l'opération est calculée par la soustraction du temps final ($temps_f$) et du temps de début ($temps_d$),
- Les différentes durées sont enregistrées dans des tableaux spécifiques à chaque opération de l'algorithme.

Les résultats obtenus sont représentés dans les Figures 4.1 et 4.2. Nous pouvons voir sur la Figure 4.1, une comparaison entre les temps consommés par les opérations : classement, mutation et croisement. Les temps consommés par l'opération évaluation sont représentés par la Figure 4.2. Nous remarquons sur la Figure 4.1 que la phase de classement consomme la plus grande partie du temps, suivie respectivement du croisement et de la mutation. Ceci peut être expliqué par le fait que la phase de classement s'effectue

sur toute la population (population + Archive) alors que les deux phases de reproduction opèrent sur une partie de la population définie par les probabilités de mutation et de croisement. De plus le croisement consomme plus de temps que la mutation car le taux d'individus à manipuler est plus grand ($P_c \geq P_m$). Cependant, les temps consommés par ces phases (classement, croisement et mutation) restent négligeables face à ceux consommés durant l'opération d'évaluation (voir 4.2). Nous obtenons une moyenne du temps consommé par la phase de classement égale à 0.3354, alors que celle de la phase d'évaluation est de 44.8189, le rapport entre ces deux durées est de 133.6282. Nous déduisons que la phase d'évaluation consomme, au minimum, 133 fois plus de temps que les autres phases de l'algorithme.

Partant de cette étude nous pouvons dire que la phase qui consomme le plus de

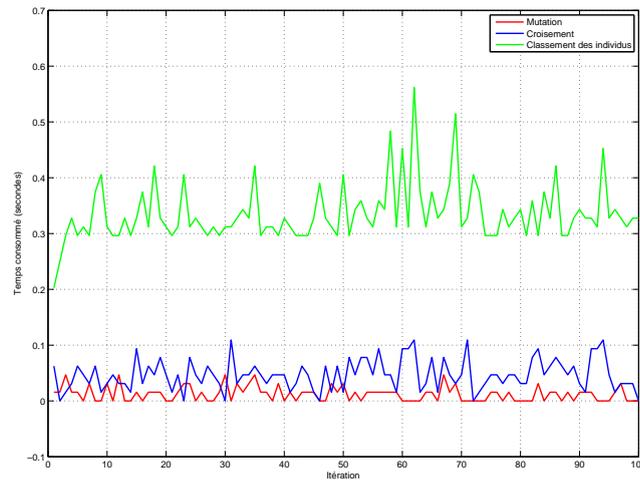


FIGURE 4.1: Temps consommé par les phases : croisement, classement et mutation

temps est l'opération d'évaluation. Cette dernière est effectuée sur tous les individus de la population, notamment ceux de l'archive pour les deux algorithmes SPEA-II, et PESA-II. Pour le reste de notre étude, nous allons donc nous intéresser de plus près à cette partie de l'algorithme. L'objectif est de trouver une solution qui limite le nombre d'évaluation des individus. Pour cela, nous nous intéressons aux algorithmes hybrides, plus spécifiquement ceux qui utilisent une notion d'approximation de la valeur de fitness. Différentes méthodes s'offrent à nous, ces dernières seront discutées en détails dans les prochaines parties.

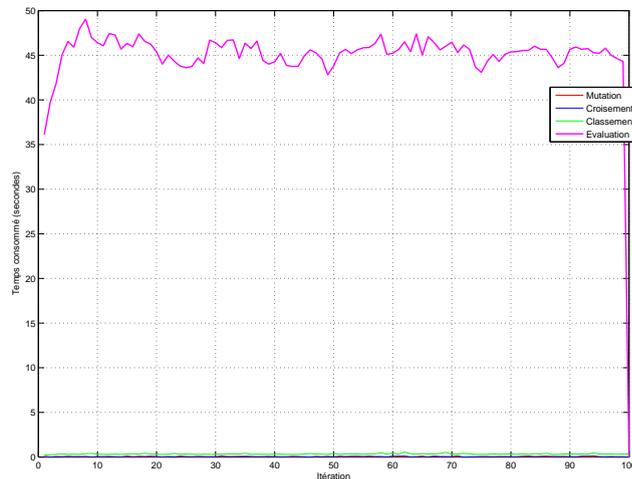


FIGURE 4.2: Temps consommé par la phase d'évaluation de l'algorithme

4.3 Hybridation dans les méta-heuristiques

Dans le dictionnaire Larousse l'hybridation est définie comme suit : *"un croisement entre deux variétés, deux races, d'une même espèce ou entre deux espèces différentes"*. En informatique l'hybridation désigne une coopération entre deux (ou plusieurs) méthodes ou algorithmes, afin de résoudre un problème spécifique [173]. Dans le domaine de l'optimisation, une hybridation de méthodes consiste à combiner plusieurs (deux au minimum) méthodes afin de tirer les avantages et améliorer les performances de l'algorithme. Ces dernières années le domaine d'hybridation de méthodes d'optimisation a connu un très grand succès dans la communauté scientifique (recherche opérationnelle) ouvrant le champs à de nouveaux horizons dans le domaine de l'optimisation combinatoire.

Selon les travaux effectués par E. G. Talbi [15] et de Gunther R. Raidl [174], les algorithmes hybrides peuvent être classés selon la problématique traitée en : problèmes de conception et problèmes d'implémentation. La première catégorie concerne l'algorithme en lui-même, incluant les problèmes de fonctionnalité et d'architecture de l'algorithme hybride développé. La seconde, concerne la plate-forme matérielle utilisée, les modèles de programmation, ainsi que, l'environnement dans lequel l'algorithme est exécuté.

4.3.1 Problèmes de conception

Selon E.-G. Talbi [15] (Figure 4.3), cette catégorie d'algorithmes hybrides peut être classés de manière hiérarchique en : hybridation à faible niveau (ou bas-niveau) et hybridation à haut-niveau. Le choix des caractéristiques peut être effectué indépendamment

du choix de conception précédent, et ainsi, la classification devient horizontale (plate). La Figure 4.3 montre la décomposition de la classe "problèmes de conception", les deux types de classification sont exposés : hiérarchique, et horizontale.

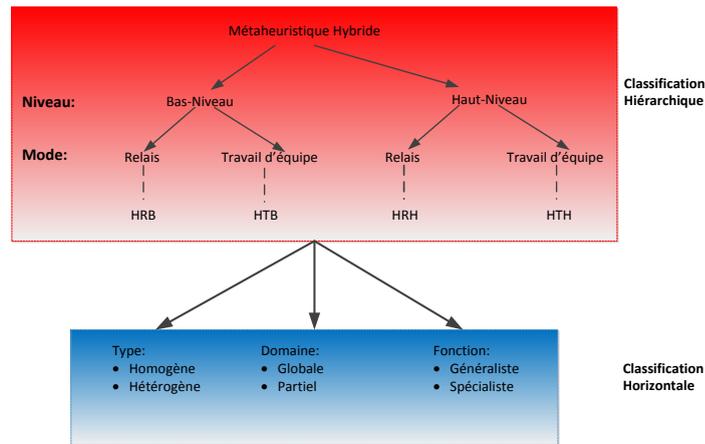
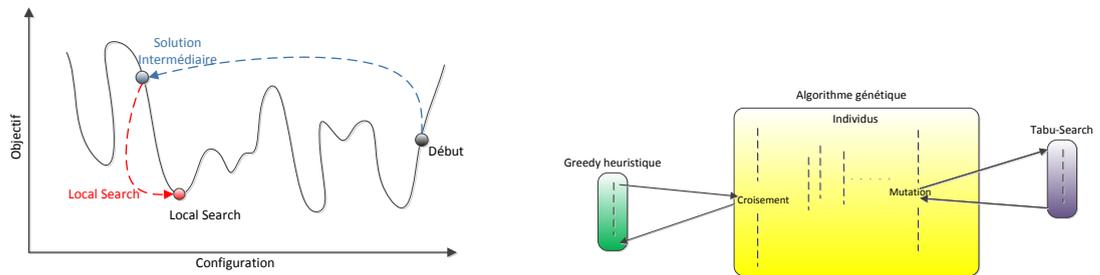


FIGURE 4.3: Classification des métaheuristiques hybrides [15]

Classification hiérarchique Dans le premier niveau nous pouvons distinguer deux sous-classes d’algorithmes : Bas-niveau et Haut-niveau. La différence entre ces deux classes réside dans la manière dont l’algorithme hybride est développé. Dans l’hybridation à bas-niveau, une fonction quelconque (ex. : mutation ou croisement dans les EA) est remplacée par une autre méta-heuristique (ex. : PSO). Dans une hybridation à haut-niveau, les méta-heuristiques sont autonomes (indépendantes). Il n’existe en aucun cas une relation directe au fonctionnement interne de ces dernières, les algorithmes combinés sont considérés comme indépendants l’un de l’autre.

Dans le deuxième niveau, on remarque deux manière de fonctionnement des algorithmes : relais, et travail d’équipe. Dans la première, les algorithmes (méta-heuristiques) sont exécutés de manière successive, où la solution trouvée par le premier algorithme constitue solution de départ (ou population initiale) du suivant. Dans la deuxième catégorie, l’hybridation est réalisée d’une manière coopérative, dans laquelle plusieurs agents coopèrent et évoluent parallèlement. Dans cette dernière, chaque agent effectue sa propre recherche dans l’espace de solution.

- *HRB (Hybridation Relai Bas-niveau)* : plus connue sous l’appellation LRH (Low-level Relay Hybrid), ce mode d’hybridation consiste à intégrer une méta-heuristique quelconque dans une S-méta-heuristique (ex. : recherche locale dans le recuit simulé). Ce type d’hybridation n’a pas été trop utilisé dans la littérature.



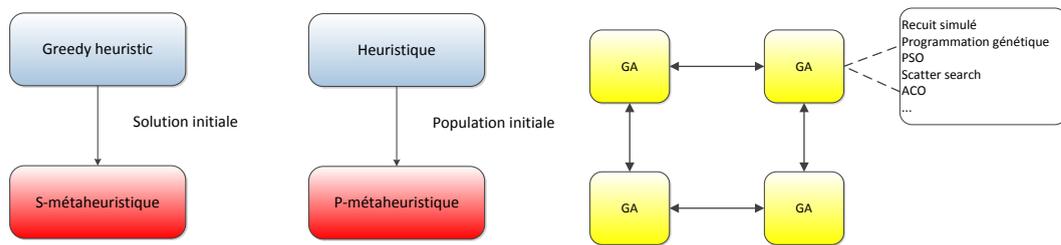
(A) exemple d'une hybridation en relais bas-niveau (HRB) (B) exemple d'une hybridation travail-d'équipe bas-niveau (HTB)

FIGURE 4.4: Architecture générale du bâtiment

- *HTB (Hybridation Travail d'équipe Bas-niveau)* : communément appelée LTH (Low-level Teamwork Hybrid), elle consiste à mettre en œuvre une stratégie de coopération entre deux méthodes d'optimisation (le plus souvent une S-métaheuristique et une P-métaheuristique) afin de tirer les avantages des deux. On rencontre souvent le problème d'exploitation et d'amélioration de la solution dans les P-métaheuristique, ainsi que, le problème d'exploration de l'espace de solutions dans les S-métaheuristique. Pour cette raison, plusieurs méthodes hybrides sont composées d'un croisement entre une P-métaheuristique et une S-métaheuristique. En effet, la S-métaheuristique va essayer d'optimiser localement, pendant que la P-métaheuristique va le faire d'une manière globale.
- *HRH (Hybridation Relai Haut-niveau)* : consiste à exécuter des algorithmes (métaheuristiques) indépendamment et de manière séquentielle. On trouve souvent l'exemple de génération de solution initiale par une heuristique (ex. : algorithme glouton) dans les S-métaheuristiques, ou bien, génération de population initiale par une heuristique à base de population dans les P-métaheuristiques (voir Figure 4.5a).
- *HTH (Hybridation travail d'équipe Haut-niveau)* : consiste à faire évoluer parallèlement et de façon indépendante deux ou plusieurs algorithmes métaheuristiques. Ce type d'hybridation permet d'avoir une stratégie d'échange d'informations entre les différents algorithmes qui opèrent indépendamment pour un seul objectif. Ceci peut minimiser le temps de calcul grâce à l'échange de solution actuelle trouvée par les algorithmes (voir Figure 4.5b).

Classification horizontale Dans cette partie, nous pouvons distinguer trois grandes familles d'algorithmes qui sont classés selon : le type d'hybridation (homogène, hétérogène), le domaine de recherche (global, partiel) et la fonction objective utilisée (générale, spécialiste).

- *Type d'hybridation* : les algorithmes hybrides peuvent être classés selon le type d'hybridation en : homogène et hétérogène. Dans la première, l'hybridation consiste à



(A) exemple d'une hybridation en relais haut-niveau (HRH) (B) exemple d'une hybridation travail-d'équipe haut-niveau (HTH)

FIGURE 4.5: Exemples d'hybridation Haut-niveau

utiliser plusieurs algorithmes du même type. On rencontre souvent une combinaison de plusieurs algorithmes génétiques qui utilisent différents paramètres (probabilités de croisement, taille de la population initiale, ...). Dans la deuxième (cas hétérogène), différentes métaheuristiques sont utilisées. On remarque souvent l'utilisation d'une S-métaheuristique telle que la recherche tabou afin de générer des solutions qui seront intégrées dans la population d'un algorithme génétique. Les meilleures solutions trouvées par la recherche tabou, forment une population d'élite pour l'algorithme génétique.

- *Domaine de recherche* : Ici, on s'intéresse plus à l'espace de recherche. Dans une hybridation de domaine globale, on utilise plusieurs métaheuristiques qui explorent le même espace de recherche. Alors que dans une hybridation de domaine partiel, l'espace est décomposé en plusieurs sous-espaces en vue d'affecter chaque sous-espace à une métaheuristique spécifique.
- *Fonction Objective* : il en existe deux types d'architecture : générale et spécialiste. Dans la première, toutes les métaheuristiques utilisées résolvent le même problème d'optimisation où on a la même fonction objective pour toutes les méthodes utilisées. Alors que dans la deuxième, plusieurs métaheuristiques sont combinées où chacune résout un problème spécifique exprimé par une fonction objective partielle (la fonction objective est divisée en plusieurs parties).

4.3.2 Problèmes d'implémentation

Ici, les algorithmes sont classés selon le matériel utilisé (environnement dans lequel l'algorithme est exécuté). Selon [175], on distingue deux types d'environnements : ordinateurs à usage général, et ordinateurs spécifiques. Dans les ordinateurs à usage général, les algorithmes sont classés selon la manière avec laquelle ils sont exécutés en : algorithmes hybrides séquentiels et algorithmes hybrides parallèles. Alors que dans les ordinateurs spécifiques, les algorithmes sont classés selon le matériel utilisé ex. : FPGA, GPU, ...

Nous nous intéressons au problème de conception et plus précisément aux architectures hybrides bas-niveau travail-d'équipe. Dans les sections suivantes nous présentons l'architecture hybride proposée. Le concept de notre algorithme consiste à intégrer de l'intelligence dans un algorithme évolutionnaire et cela par le biais de concepts des algorithmes d'apprentissage (machine-learning).

4.4 Architecture hybride proposée

Nous optons pour une approche de niveau-bas avec travail d'équipe (LTH : Low-level Teamwork Hybridation) pour traiter le problème de déploiement des RCSF. Notre choix est porté sur ce type d'hybridation, car il nous offre facilement la possibilité d'intégration de la notion d'approximation. Une approche hybride LTH consiste à utiliser des techniques de fouilles de données dans les opérateurs de re-combinaison des P-métaheuristiques. Dans cette classe de méthodes hybrides, la connaissance extraite lors de la recherche est incorporée dans les opérateurs de re-combinaison pour la génération de nouvelles solutions. En partant d'un ensemble de solutions (ex. : population actuelle), certains modèles extraits peuvent être représentés par des règles de classification, d'association, arbres de décision ou autres. Ces modèles participeront à la génération de nouvelles solutions dans le but d'intensifier ou de diversifier la recherche, ce qui est justifié par les travaux de la référence [176].

4.4.1 Approximation de la valeur de fitness

Malgré les recherches menées dans le domaine des algorithmes évolutionnaires, ces derniers souffrent de plusieurs inconvénients. Comme nous l'avons déjà mentionné (Figure 4.2 4.1), la phase d'évaluation consomme beaucoup de temps de calcul, mais aussi de la mémoire, et doit être revue. Ainsi, dans certains cas, l'utilisation de la fonction d'évaluation devient très difficile ou impossible (fonction non exprimée analytiquement) du fait de sa complexité. Dans ces cas, une approximation de la valeur de fitness doit être intégrée et adoptée dans l'algorithme d'optimisation. Selon [177], l'approximation de la valeur de fitness est généralement utilisée dans les cas suivants :

- Le temps consommé par la phase d'évaluation devient trop grand
- Impossible d'estimer la valeur réelle de fitness (changement de valeur à chaque évaluation du même individu)
- Modèle d'évaluation inexistant ou trop complexe.

Les approches d'approximation de la valeur de fitness peuvent être classées selon le problème étudié en : approximation du problème, approximation de la fonction et approximation évolutionnaire [178].

- *Approximation du problème* : On retrouve dans cette catégorie des approches visant à remplacer le problème original par un autre qui lui ressemble approximativement et qui sera plus simple à résoudre. Quand le problème traité devient très complexe ou difficile à modéliser, cette approche d'approximation est recommandée.
- *Approximation fonctionnelle* : Dans cette catégorie d'approches d'approximation la fonction objective est calculée grâce à un modèle mathématique simple. Quand la fonction objective (ou fonction de fitness dans le cas des algorithmes évolutionnaires) devient très gourmande en temps de calcul (ex. : la fonction objective est une simulation), elle peut être remplacée par un modèle mathématique simple où les entrées du modèle sont les individus à évaluer et ses sorties sont la (les) valeur(s) de la fonction objective.
- *Approximation évolutionnaire* : Cette catégorie d'approches est propre aux algorithmes évolutionnaires. Elle se divise en deux classes d'algorithmes : héritage et approximation de la valeur de fitness . Dans la première, la fonction de fitness des individus issus de la phase de reproduction est héritée à partir de la valeur de fitness de leurs parents . On retrouve plusieurs types d'estimation (d'héritage) de la valeur de fitness, nous pouvons citer comme exemple les travaux menés dans [179], où la valeur de fitness d'un enfant (solution générée par la phase de reproduction) est calculée par une somme pondérée des deux valeurs de fitness de ses parents. Dans la deuxième classe, la valeur de fitness d'un individu est calculée en se basant sur des valeurs obtenues précédemment. Il existe plusieurs modèles d'approximation tels que : Réseaux de Neurons Artificiels, régression et interpolation, modèle de krigeage,...

4.4.2 Analyse de l'existant

Dans plusieurs problèmes d'optimisation le temps de calcul et la mémoire sont des paramètres qu'on peut pas négliger. Pour cela, des algorithmes hybrides qui intègrent des méthodes de classifications sont utilisés dans le but d'approximer les fonctions objectives pendant la recherche. L'objectif est de minimiser le nombre de solutions (individus) qui seront évaluées et réduire ainsi le temps de calcul de l'étape d'évaluation des solutions. Ceci est réalisé grâce à la division de la population en plusieurs classes en utilisant des méthodes telles que les K-PPV (K-Plus Proches Voisins). Ensuite, le centre de chaque classe est évalué par la fonction objective. Les valeurs de fitness des solutions restantes sont approximées en se basant sur la valeur de la fonction objective de la classe associée (fitness imitation).

Comme nous l'avons mentionné auparavant, il existe plusieurs approches visant à réduire le nombre d'évaluations directes des individus. Dans notre cas d'étude, nous optons pour des approches d'approximation évolutionnaire. Ce type d'approche nous permet

de réduire le nombre d'évaluations directes, et ainsi réduire le temps de calcul. Ainsi, ce type d'approche est adapté aux algorithmes évolutionnaires que nous utilisons dans notre problème de déploiement des RCSF.

Selon [177, 178, 180], l'une des difficultés majeures rencontrées lors de l'application des MOEA (Multi-Objective Evolutionary Algorithms) à des problèmes de grande taille (nécessitant un grand nombre d'évaluations), est le temps de calcul consommé par la phase d'évaluation (cf. Section 4.2). En effet, l'utilisation de modèles d'approximation dans les algorithmes évolutionnaires est appréciée pour ce genre de problèmes. Ce type d'algorithmes (EA + modèle d'approximation) a reçu peu d'attention en particulier pour l'optimisation multi-objectifs. Les auteurs dans [177], proposent une revue de toutes les méthodes d'approximation de la valeur de fitness appliquées aux algorithmes évolutionnaires. Une classification des méthodes d'approximation, selon l'approche utilisée, a été proposée par les auteurs qui est comme suit : Instance Based Learning, Machine Learning et Statistical Learning. Une étude comparative entre ces différentes méthodes a été initiée dans cet article. Cette étude montre que l'utilisation de ces méthodes est efficace pour résoudre des problèmes d'optimisation multi-objectifs en peu de temps. Les auteurs rapportent que le choix de la méthode d'approximation dépend des besoins de l'utilisateur et de la complexité du problème. Ils proposent de commencer par l'utilisation de méthodes simples tels que les KPPV (K-Plus Proches Voisins) et si ces derniers ne donnent pas de bons résultats, des méthodes plus complexes tels que les RN, SVM (Support-Vector Machine), peuvent être implémentées.

D'autres travaux se sont intéressés à ces modèles afin d'approximer la valeur de fitness et diminuer le nombre d'évaluations directes. Nous pouvons citer les travaux menés par Nain et Deb [181] sur l'utilisation des RNs afin d'approximer la valeur de fitness de la population. Les auteurs proposent d'alterner entre l'évaluation directe et l'approximation de la valeur de fitness par les RNs. La méthode proposée commence par appliquer l'algorithme génétique avec une évaluation directe des individus pendant un nombre prédéfini de générations. La population issue de l'exécution d'un nombre prédéfini de génération de l'algorithme génétique constitue une base d'apprentissage du RN. Une fois le RN appris, il sera utilisé pour définir les valeurs de fitness des individus de la population pendant un nombre prédéfini d'itération. Le problème avec cette méthode réside dans la définition du bon nombre de générations pour effectuer l'alternance entre évaluation directe et approximation.

Une étude comparative entre deux méthodes d'approximation a été proposée dans [182]. Les auteurs ont mené une étude sur l'utilisation des RNs et des méthodes de krigeage pour l'approximation des valeurs de fitness dans un algorithme évolutionnaire. Les deux méthodes ont été appliquées sur les fonctions de tests suivantes : Ackley, Rosenbrock et Keane, connues dans la littérature. Deux modes d'apprentissages ont été considérés pour les deux méthodes utilisées.

Le premier mode d'apprentissage, consiste à construire un modèle d'apprentissage à partir des données issues des exécutions précédentes de l'algorithme d'optimisation.

Dans le deuxième mode, le méta-modèle est mis-à-jour à plusieurs reprises avec de nouvelles données générées par l'optimisation. Les résultats montrent que les méthodes d'approximations sont fortement recommandées pour des problèmes complexes. À partir de ces résultats, les auteurs montrent que l'utilisation de l'approximation de fitness peut dégrader la solution, surtout quand le nombre d'itération de l'algorithme est très grand.

Afin de pallier ce problème, les auteurs dans [183] proposent l'utilisation de méthodes de clustering. Au début la population initiale est générée et évaluée par la fonction objective (évaluation directe). Après cela, des solutions sont générées par le croisement et mutation de la population initiale qui seront classifiées en K clusters par l'algorithme *K-means*. Une fois la population classifiée, les K solutions les plus proches des centroïdes seront évaluées par la fonction objective (évaluation directe) et constitueront une base pour la construction du RN. Finalement, le reste de la population se voit attribuer une valeur de fitness approximative grâce au RN. La limite de cette méthode est l'apprentissage répétitif du RN qui consomme un temps de calcul assez conséquent. Aussi, l'approximation de la valeur de fitness est erronée dans certains cas.

Afin de limiter le nombre d'apprentissage du RN, et de réduire l'erreur générée par ce dernier, nous proposons une approche d'approximation de fitness par des RNs en nous basant sur les travaux [183]. Notre approche se décompose en deux parties : (1) Algorithme génétique, (2) méthodes de classification supervisées. Nous proposons l'utilisation de la méthode *K-means* (ou *K-moyennes*) pour la classification des solutions et l'élection de l'individu porteur de la classe (centroïde). Les centroïdes de chaque classe sont ensuite évalués, et les solutions restantes sont approximées en se basant sur les valeurs de fitness du centroïde de la classe associée.

Pour réduire l'erreur générée par les RN nous introduisons une fonction de fiabilité dans le calcul de la fonction objective de chaque individu. L'algorithme proposé, ainsi que la fonction de fiabilité seront présentés en détails dans les prochaines sections.

Le tableau 4.1 présente une liste (non-exhaustive) des travaux utilisant les Réseaux de Neurones pour approximer la valeur de fitness.

4.4.3 Algorithme proposé

Comme mentionné auparavant, l'algorithme proposé a pour objectif de diminuer le nombre d'évaluations directes en les remplaçant par des estimations (approximations) de la valeur de fitness des individus de la population. Nous résumons l'essentiel de notre état de l'art sur ce sujet dans le Tablea 4.1. Les différentes phases de l'algorithme

Auteurs	Année	Algorithme d'optimisation	Méthode d'approximation
A. G. Cunha et al[180]	2005	MOEA	Réseau de Neurones
Y. Jin et al[178]	2005	Genetic Algorithm	Réseau de Neurones
L. Willmes et al[182]	2003	Genetic Algorithm	Réseau de Neurones
Y. Jin et al[183]	2004	Genetic Algorithm	Clustering + Réseau de Neurones
R. G. Beiko et al[184]	2005	Genetic Algorithm	Réseau de Neurones
M. Schmidt et al[185]	1996	Gentic Algorithm	Réseau de Neurones
F. E. Cellier[186]	2013	Genetic Algorithm	Réseau de Neurones
D. Pham[187]	2012	GA, SA, tabou	Réseau de Neurones
N. Rajagopalan[188]	2012	Genetic Algorithm	Réseau de Neurones
B. Grill[189]	2015	Genetic Algorithm	Réseau de Neurones

TABLE 4.1: Travaux sur l'approximation de la valeur de fitness par des réseau de neurones artificiels

évolutionnaire utilisé (SPEA-II) restent les mêmes que celles définies dans le chapitre précédent, à savoir : génération aléatoire de la population initiale, sélection par tournoi, croisement en un point et mutation binaire. La phase d'évaluation est remplacée par deux algorithmes : (1) division (classification) de la population par la méthode *K-means* et (2) estimation (approximation) de la valeur de fitness par les *Réseaux de Neurones*. Une fonction de fiabilité de la solution est développée afin de minimiser l'erreur introduite par les RNs. Grâce à cette fonction, la valeur de fitness d'un individu est calculée par la combinaison des deux valeurs de fitness : estimée (RN) et valeurs des centroides (K-means). Ce méta-modèle a été validé par [190]. Les auteurs ont testés le méta-modèle sur les différentes fonctions complexes connues (Rastrigin, Rosenbrock, ...). Les résultats montrent que l'association d'une méthode d'approximation telle que les réseaux de neurones et des méthodes de clustering diminue le temps de calcul (évaluation directe) tout en assurant une très bonne qualité de la solution. L'algorithme proposé par les auteurs a été testé sur des problèmes mono-objectif. Au meilleur de notre connaissance, aucun travail associant RN et méthode de clustering n'a été proposé pour des problèmes multi-objectifs.

Nous proposons dans notre méthode, l'adaptation du méta-modèle proposé dans les travaux de Yoon et al. dans [190] à l'optimisation multi-objectifs. Cela, devrait être assuré par la notion de dominance qu'intègre la méthode SPEA-II. L'algorithme que nous proposons se déroule comme suit :

1. *Génération de la population initiale* : Cette phase est assurée par une méthode aléatoire afin de générer des solutions de départ qui constitueront la population initiale de l'algorithme SPEA-II.

2. *Estimation de la valeur de fitness* : Cette phase constitue le cœur de notre travail, elle se divise en deux parties distinctes. Dans la première partie, la population est divisée en plusieurs classes grâce à la méthode *K-means*. Une fois la population classifiée, les individus les plus proches des centroïdes seront évalués directement (avec la fonction objectif). Le reste de la population se voit attribuer une valeur de fitness approximative qui sera représentée par une combinaison des deux valeurs de fitness du centroïde et celle approximée par le RN.
3. *Classement de la population* : La méthode de classement des individus de la population reste la même que celle utilisée dans le chapitre précédent (SPEA-II).
4. *Sélection et reproduction* : Une sélection par tournoi est utilisée (la même que dans le chapitre 3). Pour ce qui est de la reproduction (croisement et mutation), la méthode ne change pas, sauf lors de l'évaluation des nouveaux individus.

La Figure 4.6, représente la démarche suivie par l'algorithme proposé. Vu que les phases de génération de la population initiale, sélection et reproduction n'ont pas changé, nous nous limiterons dans les sections suivantes qu'aux étapes rajoutées.

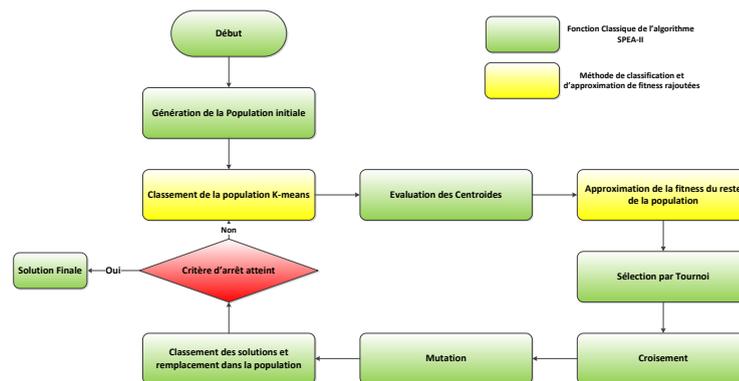


FIGURE 4.6: Architecture de l'algorithme proposé

4.4.4 Limiter les évaluations

Deux méthodes ont été intégrées à l'algorithme SPEA-II, afin de limiter les évaluations. Ceci est assuré grâce à l'algorithme *K-means* qui divise la population en plusieurs classes $C_k = c_1, c_2, \dots, c_K$, où K représente le nombre de classes identifiées par l'algorithme *K-means*. Dans chaque classe un centroïde (centre de la classe) est défini par l'algorithme. Une fois les centroïdes définis, nous procédons à l'identification des individus les plus proches. Ces individus (les plus proches des centroïdes) seront évalués directement avec la fonction objective. Les valeurs de fitness des autres individus de la population sont

approximées en utilisant les Réseaux de Neurones et la valeur de fitness des centres de classes les plus proches.

4.4.4.1 Algorithme *K-means*

Depuis plusieurs années, l'algorithme *K-means* reste le plus utilisé pour la classification de données [191]. Cet algorithme est connu pour ses nombreux avantages dont sa facilité d'implémentation, sa simplicité ainsi que son efficacité. La méthode *K-means* peut être définie comme suit :

Soit l'ensemble $X = x_i, i = 1, \dots, n$ un ensemble de points dans un espace de n -dimensions, qu'on veut classer en K classes, $C = c_k, k = 1, \dots, K$. L'algorithme *K-means* essaiera de trouver la bonne partition telle que l'erreur quadratique entre le centre empirique de la classe et tous les points lui appartenant soit minimale. L'erreur quadratique entre un centre μ_k de la classe c_k et tous les points appartenant à c_k , est définie comme suit :

$$E(c_k) = \sum_{x_i \in c_k} (\|x_i - \mu_k\|^2) \quad (4.1)$$

Le but de l'algorithme est de minimiser la somme des erreurs quadratiques sur toutes les classes K . Cette somme est exprimée comme suit :

$$E(C) = \sum_{k=1}^K \left(\sum_{x_i \in c_k} (\|x_i - \mu_k\|^2) \right) \quad (4.2)$$

Selon [192], la minimisation de cette somme est un problème connu pour être *NP-Difficile* (même quand $K = 2$). Aussi, les auteurs précisent que l'algorithme *K-means* risque converger vers un optimum local, mais assure des délais d'exécution très courts. L'algorithme se déroule comme suit :

1. Sélectionner une partition avec les K -classes ;
2. Générer une nouvelle partition en affectant chaque point au plus proche centre de classe ;
3. Calculer de nouveaux centres de classes ;
4. répéter les étapes 2 et 3 jusqu'à ce que les membres de la classe se stabilisent.

Dans notre algorithme, le nombre de classes dépend du nombre de fonctions objectives. Pour ce qui est de la phase de déploiement des nœuds routeurs le nombre de classes est égale à 2, ce qui représente les objectifs : coût et connectivité. Lors de la phase de déploiement des nœuds capteurs, le nombre de classe est fixé à cinq classes, ce qui correspond aux différents objectifs : coût, connectivité, couverture, sur-couverture et durée de vie.

Grâce à ce choix du nombre de classes, les individus de la population sont regroupés selon la valeur de leurs fonctions objectives. L'algorithme *K-means* nous permet, par la suite d'effectuer une recherche locale de la solution qui représente le mieux l'ensemble des individus de la même classe (centroïde).

Une fois cette phase effectuée et le centroïde définit les valeurs de fitness du reste des individus de la classe seront approximées avec les réseaux de neurones artificiels. Cette phase sera présentée dans la section suivante.

4.4.4.2 Réseau de Neurones

Le principe général des Réseaux de Neurones Artificiels (RNA) est à l'origine inspiré de certaines fonctions de base des neurones naturels du cerveau [193] (Figure 4.7). Ces derniers sont apparus dans les années cinquante avec les premiers perceptrons, et sont utilisés industriellement depuis les années quatre-vingt [194].

Les réseaux de neurones sont des outils très utilisés pour la classification, l'estimation, la prédiction et la segmentation. Ils sont issus de modèles bio-inspirés et sont constitués d'unités élémentaires (les neurones) organisées selon une architecture spécifique [194]. Les réseaux de neurones artificiels sont des modèles de calcul dont la conception est schématiquement inspirée du fonctionnement cérébral et des structures neuronales. Ils sont généralement optimisés par des méthodes d'apprentissage de type statistique qui leur permettent de prendre des décisions en s'appuyant davantage sur la perception que sur le raisonnement logique formel [195].

Un réseau de neurones artificiel est constitué de plusieurs neurones connectés via des

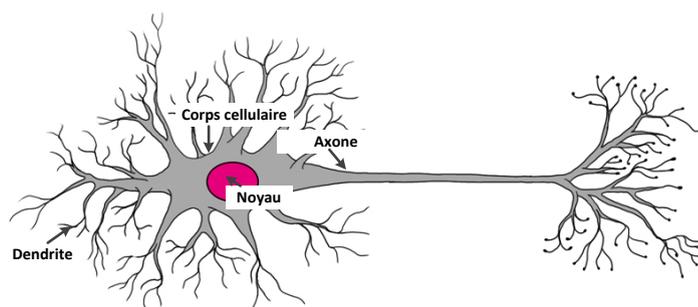


FIGURE 4.7: Schéma d'un neurone biologique (adapté de [16])

liaisons leur permettant d'envoyer et de recevoir des signaux en provenance des neurones qui les précèdent. Chacune de ces connexions reçoit une pondération W_i qui détermine son impact sur les neurones qu'elle connecte. Chaque neurone dispose ainsi d'une entrée, qui lui permet de recevoir de l'information d'autres neurones, mais aussi d'une fonction

d'activation φ et enfin d'une sortie comme le montre la Figure 4.8.

Selon , un neurone artificiel est constitué d'un intégrateur effectuant la somme pondérée

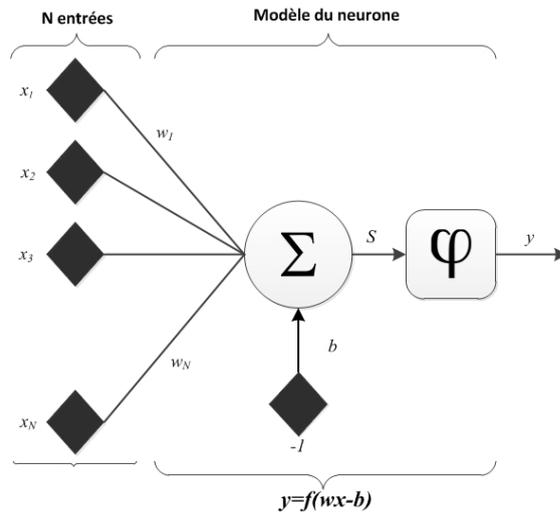


FIGURE 4.8: Modèle d'un neurone artificiel [17]

de toutes les entrées. Le résultat est ensuite calculé par la fonction de transfert (d'activation) φ qui fournit la sortie du neurone notée y . L'intégrateur délivrant la somme pondérée (S) des entrées X_i peut être décrit comme suit :

$$S = \sum_{i=1}^N (\omega_i x_i - b) \quad (4.3)$$

Le résultat S de la somme pondérée des entrées X_i est appelé "niveau d'activation". Ce niveau varie en fonction de la valeur du seuil b (appelé aussi, biais du neurone). La fonction d'activation du neurone (f) peut, alors, être représentée par l'équation suivante :

$$y = f(S) = f\left(\sum_{i=1}^N (\omega_i x_i - b)\right) \quad (4.4)$$

Il existe plusieurs fonctions d'activation, les plus utilisées restent : seuil et linéaire. Les fonctions d'activation les plus connues sont comme suit :

– *seuil* : l'équation qui définit cette fonction est comme suit :

$$y = \begin{cases} 0 & \text{si } S < 0 \\ 1 & \text{si } S \geq 0 \end{cases} \quad (4.5)$$

Il existe plusieurs variantes de cette fonction telle que le seuil symétrique.

– *linéaire* : l'équation qui définit cette fonction est comme suit :

$$y = S \quad (4.6)$$

Il existe plusieurs variantes de cette fonctions telles que : linéaire saturée, linéaire saturée symétrique et linéaire positive.

- *sigmoïde* : l'équation qui définit cette fonction est comme suit :

$$y = \frac{1}{1 + e^{-s}} \quad (4.7)$$

- *tangente hyperbolique* :

$$y = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (4.8)$$

4.4.4.3 Combinaison de la fonction de fitness

Le calcul de la valeur de fitness des individus de la population est assuré par l'équation suivante :

$$F_k(ind_i) = \rho_{ki}F_R(c_k) + (1 - \rho_{ki})F_{RN}(ind_i) \quad (4.9)$$

Où

- $F_k(ind_i)$ représente la fonction de fitness de l'individu ind_i de la classe (cluster) k
- F_R et F_{RN} représentent respectivement la fonction de fitness originale (réelle) et la fonction de fitness calculée par le RN
- c_k représente l'individu le plus proche du centroïde de la classe k
- ρ_{ki} est une fonction de fiabilité de la valeur de fitness de l'individu ind_i par rapport au centroïde c_k

La fonction de fiabilité de la solution est calculée comme suit :

$$\rho_{ki} = \frac{1}{e^{\sigma(DR_{ki})}} \quad (4.10)$$

DR_{ki} représente le ratio entre la distance euclidienne ($d(ind_i, c_k)$), et la distance entre le centroïde c_k et l'individu ind_j le plus éloigné. Ce paramètre est défini comme suit :

$$DR_{ki} = \frac{d(ind_i, c_k)}{\max_j(d(ind_j, c_k))} \quad (4.11)$$

4.5 Résultats et discussion

Afin de conduire une étude comparative entre la méthode proposée et les méthodes mentionnées dans le chapitre 3, nous reprenons les mêmes expérimentations. Ces dernières se divisent en deux parties : grande et petite instance. Les environnements de déploiements cités dans le chapitre précédent sont repris afin de conduire l'étude.

4.5.1 Expérimentation-I (Petite Instance)

Dans un premier temps nous évaluons la méthode proposée dans un environnement restreint. Comme mentionné dans le chapitre précédent, l'espace de déploiement représente une partie du bâtiment du CESI de Rouen sur une surface totale de $100m^2$. Les principaux paramètres utilisés sont les mêmes que ceux présentés dans le chapitre précédent, à savoir : taille de la population, probabilités de croisement et mutation, paramètres capteurs/routeurs, ...

La Figure 4.10 (déploiement des nœuds capteurs) et la Figure 4.11 (déploiement des nœuds routeurs) montrent la convergence de tous les algorithmes étudiés durant leurs exécutions. Nous pouvons voir sur le graphique 4.11 que la méthode proposée surpasse les autres avec une valeur de fitness $f = -86$ (rappel : $f = \omega_1 \text{Coût} + \omega_2 \text{CnC}$) dans le cas du déploiement des nœuds routeurs. Cette solution correspond à 3 nœuds routeurs déployés assurant une connectivité de 100 %. Nous pensons que ceci est dû à l'erreur introduite par les réseaux de neurones durant l'estimation de la valeur de fitness des individus de la population. Cette erreur pourrait influencer le choix des individus restant dans la population lors de la phase de remplacement. Nous gardons ainsi des individus qui étaient voués à disparaître de la population et créer une diversité. Dans le cas du déploiement des nœuds capteurs, notre méthode converge autant que les autres vers la valeur de fitness $f = -45.8$ qui correspond à dix (10) capteurs déployés.

La Figure 4.12 représente l'évolution de la valeur des différentes fonctions objectives (cas du déploiement des nœuds capteurs) durant l'exécution de l'algorithme proposé. Ces fonctions ont été normalisées afin de permettre leur représentation sur un seul et même graphique. Les fonctions de coût et de sur-couverture (espace couvert par plusieurs nœuds) sont minimisées durant chaque génération de l'algorithme (Figure 4.12). La fonction de durée de vie, quant à elle, est maximisée durant l'exécution de l'algorithme. Les deux fonctions de couverture et de connectivité sont stables à la valeur 0.042 qui correspond à 100% de couverture et de connectivité. Ceci est dû au grand nombre de nœuds déployés au début de l'algorithme qui couvrent tout l'espace (avec des parties sur-couvertes par plusieurs nœuds) et assure une connectivité dans n'importe quel endroit de l'espace de déploiement. Ce n'est que vers la fin de l'algorithme que la valeur de ces deux fonctions est vu à la baisse, ce qui correspond à un nombre de nœuds très faible.

Les deux Figures 4.13 et 4.14 montrent le temps consommé dans chaque génération des algorithmes étudiés. Nous remarquons que grâce à l'élimination partielle de la phase d'évaluation directe (remplacée par l'approximation de la valeur de fitness), la méthode proposée consomme moins de temps en assurant les mêmes performances que les autres méthodes. En effet, la méthode d'approximation proposée permet de diviser en deux le

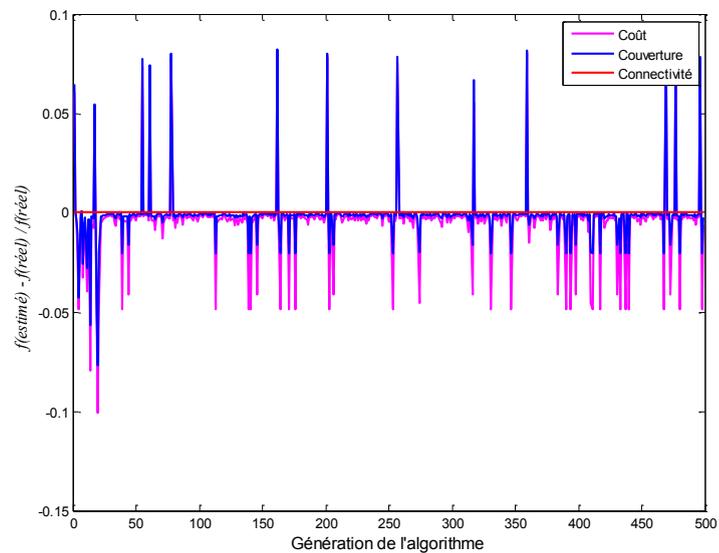


FIGURE 4.9: Erreur quadratique entre les valeurs de fitness estimée et réelle

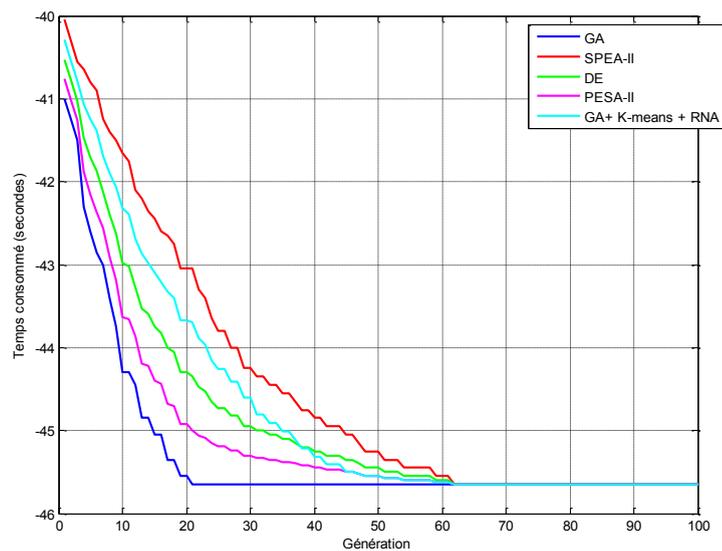


FIGURE 4.10: valeur de fitness vs génération de l'algorithme(déploiement capteurs)

temps consommé. Nous remarquons par contre un léger retard au début de l'exécution de l'algorithme par rapport aux autres. Ceci est dû à la phase d'apprentissage du réseau de neurones lors de la première itération. Malgré ce retard, nous pouvons dire que la méthode proposée reste la plus performante.

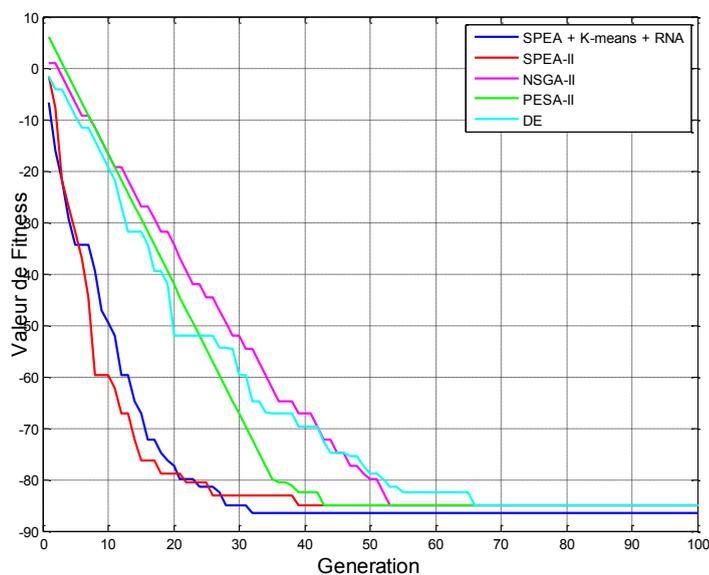


FIGURE 4.11: valeur de fitness vs génération de l'algorithme (déploiement routeurs)

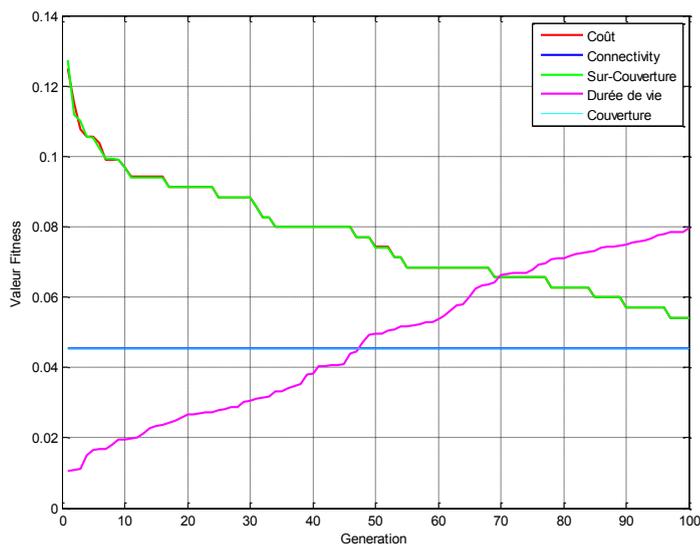


FIGURE 4.12: Fonctions objectives normalisées

4.5.2 Expérimentation-II (Grande Instance)

Une fois les performances de la méthode proposée validées lors du test sur un espace de déploiement restreint, nous avons procédé au test sur le bâtiment étudié dans le chapitre précédent. L'espace sélectionné représente la partie recherche du CESI de Rouen qui s'étend sur une surface de $850m^2$. L'immeuble comporte plusieurs salles disposées

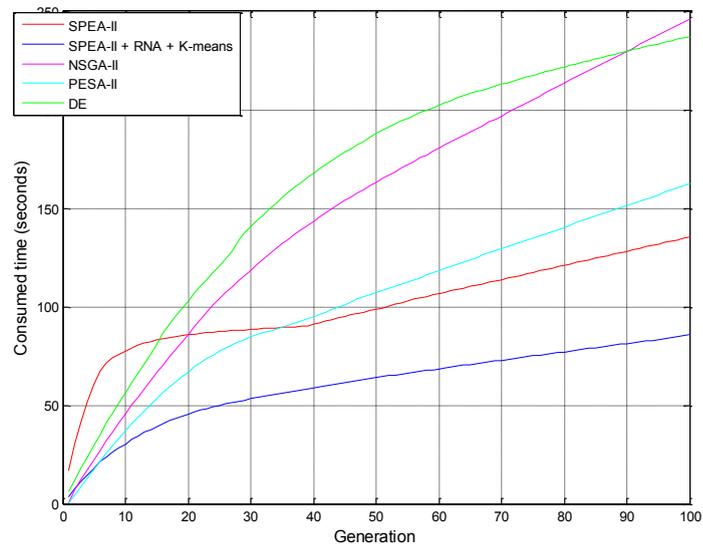


FIGURE 4.13: valeur de fitness vs génération de l'algorithme

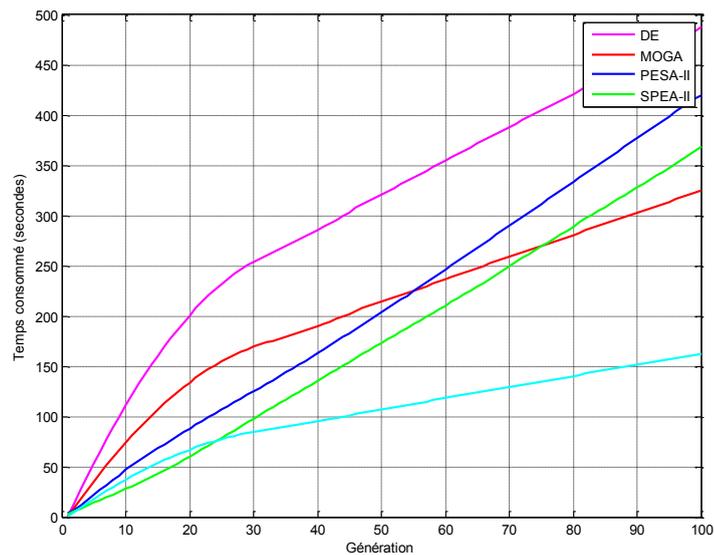


FIGURE 4.14: Temps cumulé dans chaque génération de l'algorithme (déploiement capteurs)

sur deux étages.

Afin de pouvoir intégrer le déploiement sur deux étages, nous améliorons le simulateur de propagation d'onde, ainsi que celui de la couverture, qui permettent de calculer les valeurs de couverture et de connectivité. Le simulateur pourra ainsi calculer la propagation d'ondes sur un espace 3D et prendra en charge les atténuations liées aux séparations entre les deux étages. Dans cette étude nous appliquons notre méthode d'optimisation

hybride sur le premier étage en considérant la hauteur des murs.

La Figure 4.15 et la Figure 4.16 représentent une étude comparative entre la méthode proposée et les méthodes étudiées dans le chapitre précédent (Phase de déploiement des nœuds routeurs). Nous pouvons voir sur la Figure 4.15 la méthode proposée assure une bonne convergence vers la solution trouvée par les autres méthodes. La solution trouvée consiste en un déploiement de treize (13) nœuds routeurs assurant une connectivité dans toutes les zones du bâtiment. Nous remarquons aussi que l'algorithme proposé ne converge pas rapidement vers la solution et reste dans des minima locaux durant plusieurs générations. Ceci est dû à la grande dimension de l'espace de recherche (850 variables). L'estimation de la valeur de fitness devient parfois impossible car les individus ne sont plus reconnus car le RN n'est appris qu'avec les individus de la première génération. Du coup, il existe plusieurs parties de l'espace de recherche que le réseau de neurones ne peut reconnaître ce qui introduit une erreur lors de l'estimation de la valeur de fitness. Néanmoins, notre algorithme réussit à trouver la bonne valeur de fitness après un certain nombre de génération.

Pour ce qui est du temps consommé, notre algorithme affiche de meilleures performances en divisant le temps consommé par cinq (05) fois. Ceci explique le choix de l'estimation de la valeur de fitness des individus.

Pour ce qui est du déploiement des nœuds capteurs, notre algorithme arrive à trouver une solution moins bonne que les autres méthodes étudiées. Ceci est dû à l'erreur engendrée par la phase d'approximation de la valeur de fitness (réseau de neurones). Nous avons remarqué qu'à partir d'un certain nombre d'itérations l'estimateur utilisé introduit une erreur lors du calcul de la valeur de fitness. Ceci élimine ou garde plusieurs individus dont la valeur de fitness a été faussement approximée.

Par contre, en analysant les temps consommés (Figure 4.17) par les différents algorithmes étudiés, nous remarquons que notre méthode reste la plus performante. L'estimation de la valeur de fitness permet un gain considérable du temps de calcul malgré ses inconvénients.

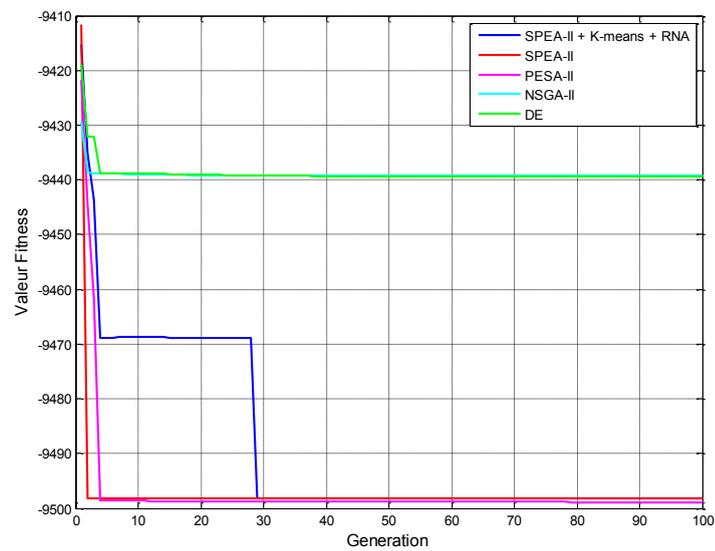


FIGURE 4.15: Valeur de fitness vs génération de l'algorithme

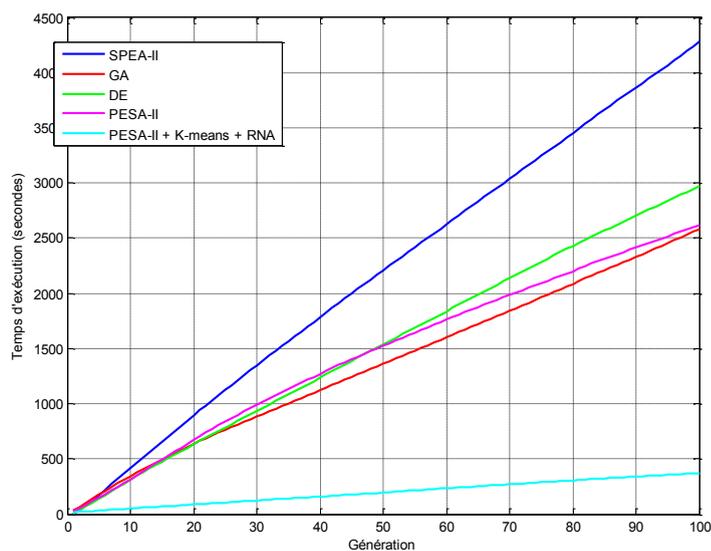


FIGURE 4.16: Temps cumulé par génération de l'algorithme

4.6 Outil de déploiement des RCSF

Afin d'afficher le résultat final obtenu par la méthode d'optimisation proposée, nous avons développé un outil doté d'une interface graphique sous MATLAB. Le menu principal de l'outil (représenté dans la Figure 4.18) est constitué de quatre boutons définis comme suit :

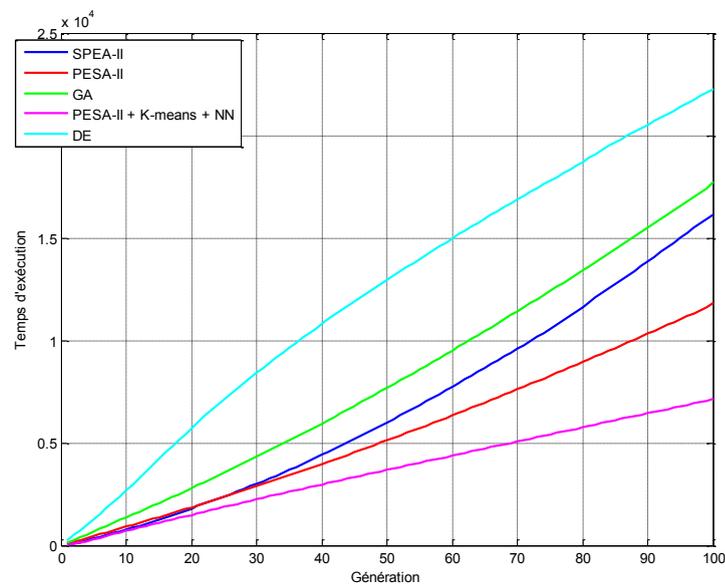


FIGURE 4.17: Temps cumulé par génération de l'algorithme (déploiement capteurs)



FIGURE 4.18: Interface principale de l'outil de déploiement des RCSF

1. Novice : ce bouton peut être utilisé par un utilisateur ne disposant d'aucune connaissance dans le domaine de déploiement des RCSF. Dans ce cas, le déploiement est totalement assuré par l'outil sans aucune intervention de l'utilisateur. L'outil calcule le nombre et les positions et affiche à la fin le résultat obtenu.
2. Expert : ce bouton offre la possibilité de tester une configuration réseau existante. Dans ce cas, l'outil joue un rôle de simulateur de : connectivité, couverture ou durée de vie.
3. Aide : ce bouton permet l'affichage du menu d'aide à l'utilisation de l'outil proposé.
4. Quitter : ce bouton permet la fermeture définitive de l'outil.

Une fois la solution trouvée, l'outil affiche les résultats (positions des nœuds) dans le plan 3D du bâtiment. Sur la Figure 4.19 nous pouvons voir une vue 3D de la configuration réseau obtenue dans l'expérimentation-II (cas déploiement routeurs). Sur cette fenêtre, nous pouvons déduire le nombre de nœuds déployés, leurs positions respectives ainsi que les taux de connectivité, couverture. Deux curseurs disposés à droite du plan du bâtiment permettent d'effectuer un balayage de l'espace de déploiement afin d'avoir les puissances de signal dans chaque cellule (balayage horizontal et vertical).

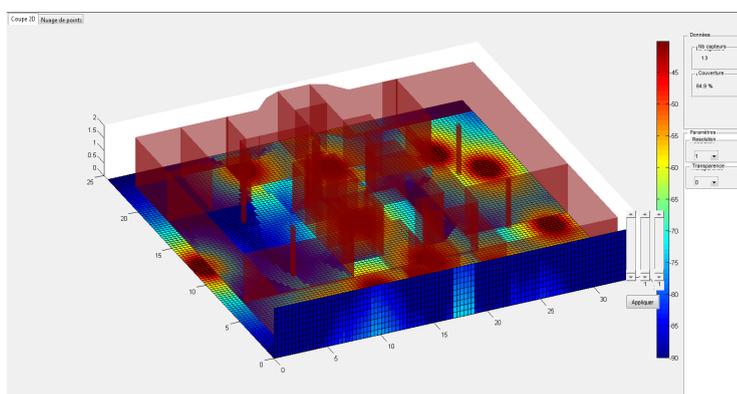


FIGURE 4.19: Vue 3D de la propagation radio

La Figure 4.20 affiche la configuration réseau obtenue dans l'expérimentation-II (cas déploiement des nœuds capteurs). Le nombre ainsi que les taux de couverture des événements sont exposés dans la figure. Le mode d'affichage est représenté sous forme de nuage de points, où chaque point représente le pourcentage (%) de couverture dans la cellule. Le nombre de nœuds ainsi que le taux totale de couverture d'événements (de tous les nœuds capteurs déployés) est fournit dans l'onglet encadré à droite du plan du bâtiment.

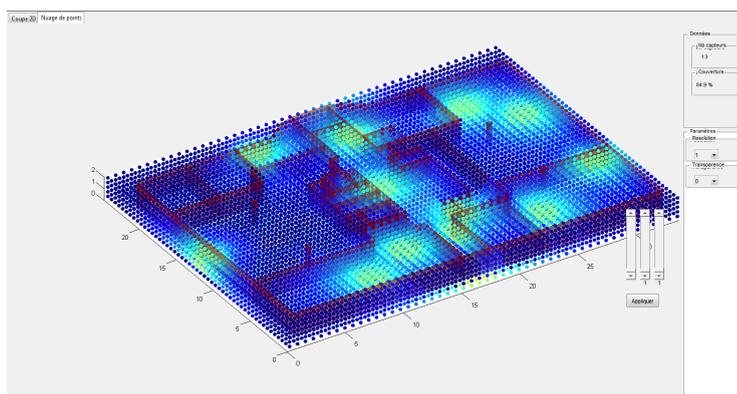


FIGURE 4.20: Vue 3D de la couverture des capteurs déployés

4.7 Conclusion

Dans ce chapitre nous avons commencé par une présentation des causes qui mènent à une lenteur des méthodes classiques. Nous avons identifié la phase d'évaluation qui consomme énormément de temps de calcul. Afin de répondre à cela, nous avons présenté les différentes méthodes d'optimisation hybrides permettant d'améliorer les performances des méthodes classiques. Un état de l'art complet de ces méthodes a été présenté dans la première partie de ce chapitre. Ensuite, nous avons présenté l'architecture de la méthode d'hybridation proposée. Cette méthode s'appuie sur des notions de clustering et d'analyse de données afin de réduire le temps consommé dans chaque itération de l'algorithme. Ceci est assuré grâce à l'estimation de la valeur de fitness en se basant sur un réseau de neurones appris avec la première population. Nous avons remarqué au début que l'utilisation du réseau de neurones seule ne permet pas une bonne convergence de l'algorithme car elle introduit une erreur lors de l'approximation de la valeur de fitness. Deux techniques de réduction de l'erreur ont été présentées. La première se base sur le ré-apprentissage du réseau de neurones dès qu'une nouvelle solution est générée. La deuxième se base sur des notions de clustering (*k-means*) afin de diviser la population en sous-parties et réguler l'erreur en évaluant les centroides. Nous avons choisi d'implémenter la deuxième technique, car elle permet un gain de temps (en évitant de passer par une phase d'apprentissage à chaque fois). Aussi, cette méthode n'a pas de contraintes de ré-apprentissage donc pas de nombre de génération à définir. En revanche, la méthode *K-means* nécessite un nombre de classes prédéfinies.

Les résultats obtenus valident notre choix, en effet, la méthode d'optimisation hybride proposée permet de réduire le temps de calcul dans les deux cas étudiés (petite et grande instance). Elle évite de lancer plusieurs simulations (de propagation d'ondes et de couverture) dans chaque itération de l'algorithme. Son seul inconvénient reste l'erreur introduite par le réseau de neurones, qui parfois bloque l'algorithme dans une solution. Finalement, nous pouvons dire que notre méthode représente un bon rapport temps/qualité (de la solution). L'approche proposée pourrait être un point de départ (une base) pour les concepteurs de réseaux de capteurs sans fil ouverte aux améliorations. Nous avons intégré cette méthode dans l'outil de déploiement présenté, nous offrant une vue globale et 3D sur le réseau déployé.

Chapitre 5

Conclusion et Perspectives

Conclusion

Dans cette thèse, nous avons proposé une nouvelle méthode d'optimisation du déploiement de RCSF pour des applications de type bâtiment intelligent. Cette méthode repose sur plusieurs modèles et simulateurs afin de calculer les différentes métriques : coût, connectivité, couverture, durée de vie et tolérance aux pannes. Ce calcul permet de trouver, au mieux, une solution qui s'adapte aux besoins définis par l'utilisateur selon plusieurs objectifs. Un outil de déploiement a été développé afin d'assister les concepteurs de RCSF durant les phases de construction et d'évaluation du réseau.

Les modèles proposés en termes de connectivité et de couverture permettent d'automatiser le calcul et ainsi libérer les concepteurs des approches classiques qui consomment énormément de temps et de ressources. Un algorithme permettant la reconstruction du plan du bâtiment en 3D à partir d'image 2D a été intégré à l'outil afin de faciliter le déploiement du RCSF sur plusieurs étages.

Ce plan 3D est utilisé par les deux simulateurs de connectivité et de couverture afin de calculer ces métriques. Le simulateur de propagation d'ondes permet de calculer la trajectoire et la puissance de l'onde en prenant en compte les atténuations provoquées par les différents obstacles présents dans le bâtiment. Pour ce faire, nous nous sommes basés sur le modèle MWF (Multi-Wall and Floor) de propagation que nous avons adapté la prise en compte des phénomènes de réfraction et diffraction de l'onde. Nous avons démontré que ce modèle est le mieux adapté dans ce type d'application de déploiement assisté par ordinateur car il permet de calculer efficacement la trajectoire de l'onde en un minimum de temps. Aussi, nous avons intégré dans le même simulateur un modèle permettant de calculer le pourcentage de zones couvertes par les nœuds capteurs déployés (couverture des événements). Ce modèle s'adapte à plusieurs types de nœuds capteurs

(température, humidité, présence,...) en prenant en compte tous les obstacles (murs, fenêtres, ...), ce qui permet de couvrir le plus d'applications possibles. Une campagne de mesures a été menée afin de définir les atténuations provoquées par les différents obstacles (murs, fenêtres, portes,...). Cette expérimentation nous a permis de définir les atténuations des matériaux présents dans le bâtiment et a validé le choix du modèle de propagation. En ce qui concerne la durée de vie, nous avons développé un simulateur qui se base sur le protocole LEACH afin de calculer la consommation énergétique du réseau et ainsi trouver une solution qui la minimise.

Après avoir étudié la modélisation d'un RCSF, nous avons vu que le problème de déploiement de ces derniers représente un problème d'optimisation Multi-Objectifs. Afin de le résoudre nous avons étudié plusieurs méthodes de résolution de ce type de problèmes. La complexité de ce dernier étant NP-Difficile nous nous sommes limités pour la suite aux méta-heuristiques et plus spécialement algorithmes évolutionnaires. Nous avons identifié six méthodes dont quatre (04) Multi-Objectifs et deux (02) Mono-Objectif. Ces méthodes sont comme suit : MOGA, NSGA-II, SPEA-II, PESA-II et les Mono-Objectifs : GA, et DE. Afin d'appliquer ces méthodes à notre problème de déploiement des RCSFs une étape d'adaptation a été nécessaire. Nous avons en premier lieu, proposé un codage simple de la solution qui combine nombre de nœuds déployés et leurs positions respectives. Ce codage peut être utilisé dans les deux cas (déploiement capteurs et routeurs) et minimise l'espace mémoire consommé en utilisant des variables binaires. Ensuite, nous avons mené une série d'expérimentations afin de définir les principaux paramètres des algorithmes étudiés, à savoir : probabilités de mutation et de croisement, taille de la population et critère d'arrêt (nombre de générations). Une fois ces paramètres identifiés nous avons procédé à une étude comparative sur deux cas réels. Les résultats obtenus nous ont permis d'identifier les meilleures méthodes à utiliser dans chaque cas d'étude (petite et grande instance). Nous avons aussi remarqué que ces algorithmes restent limités quand la taille du problème devient très importante.

Afin de remédier à cela, nous avons proposé une méthode d'optimisation Multi-Objectifs hybride. Nous avons commencé par observer et étudier les temps consommés par chaque opération des algorithmes durant toutes les générations. Nous avons remarqué que la phase d'évaluation est la plus gourmande en terme de temps de calcul. Ceci confirme notre hypothèse, car cette phase utilise plusieurs simulations afin de calculer la valeur des différents objectifs. Afin de limiter ces simulations, nous avons proposé d'utiliser des méthodes d'apprentissage pour l'approximation de la fonction de fitness. Ceci évite de recourir aux simulations trop coûteuses en temps de calcul. Cette méthode combine les notions de RNA (Réseau de Neurones Artificiels) et *K-means* afin d'identifier des individus représentant un sous-ensemble de la population (centroïdes) qui seront évalués par simulation. Les valeurs de fitness du reste de la population sont ensuite approximées grâce aux RNA. Les résultats obtenus varient selon la taille du problème. Sur des petites

instances, notre algorithme affiche de meilleures performances en termes de temps de calcul et de qualité de la solution. Par contre, dès que la taille du problème augmente (test sur une grande instance) notre algorithme n'atteint pas toujours la valeur souhaitée mais représente un gain de temps considérable. Notre méthode est donc un bon rapport entre la qualité de la solution et le temps de calcul nécessaire.

Enfin, toutes ces contributions ont été intégrées dans l'outil développé. Cet outil dispose d'une interface graphique simple à utiliser. Elle permet d'importer des plans de bâtiment 2D sous format dxf afin de générer un plan 3D. Ensuite, à l'aide des simulateurs et de la méthode d'optimisation Multi-Objectifs proposée, l'outil définit le nombre ainsi que les positions respectives des différents nœuds formant le réseau. Le résultat est affiché sous forme de vue 3D globale du bâtiment et nœuds déployés. L'utilisateur pourra ainsi voir les positions et les métriques correspondant à ces besoins.

Perspectives

Dans le futur, ce travail ouvre des perspectives de développement à plusieurs sujets de recherche :

Actuellement nous sommes en train de travailler sur la tolérance aux pannes qui est une importante facette des challenges que présente un RCSF. Afin de définir une méthode permettant de calculer la tolérance aux pannes de notre RCSF nous nous intéressons au coefficient de clustering (ou connexité) d'un graphe. Ce coefficient nous permet de calculer la probabilité qu'un lien existe entre deux voisins, tirés au hasard d'un nœud s . Selon [196], le coefficient de clustering d'un nœud v d'un graphe $G = (V, E)$ possédant plusieurs de nœuds voisins regroupés dans l'ensemble $\nu(v)$ ($\nu(v) = u \in V : \exists(u, v) \in E$), est défini comme suit :

$$C_v = \frac{(|(u, k) \in E : u \in \nu(v) \ k \in \nu(v)|)}{|\nu(v)|} \quad (5.1)$$

Ensuite le coefficient de clustering global C_G (du réseau ou graphe G) est donné par la moyenne des coefficients de tous les nœuds lui appartenant. Ce coefficient est calculé selon l'équation suivante :

$$C_G = \frac{1}{N} \sum_{v \in V} C_v \quad (5.2)$$

Ce coefficient a été largement utilisé dans le domaine des réseaux afin de calculer le degré de formation de cliques (sous-ensembles du graphe G) au sein d'un réseau. Concernant la tolérance aux pannes, nous pouvons citer les travaux menés par Matthias R. Brust et al. [197], qui ont travaillé sur la possibilité de calculer la tolérance aux pannes des RCSF grâce au coefficient de clustering. Les résultats trouvés montrent qu'un réseau disposant

d'un grand coefficient de clustering est plus tolérant aux pannes soudaines. Pour cela, ce paramètre est intégré comme fonction objectif indirecte afin d'améliorer la tolérance aux pannes du réseau déployé.

D'autres perspectives de recherche portent sur la maquette numérique du bâtiment. Ainsi, le plan de bâtiment 2D (format dxf) pourrait être remplacé par des modèles BIM qui offrent plusieurs informations sur le bâtiment. Ces informations peuvent être traduites par la suite en contraintes de placement ou pourraient constituer des supports pour le calcul du coût de maintenance du RCSF.

D'un autre côté, l'outil pourrait intégrer le déploiement de plusieurs types de capteurs avec des applications différentes. Aussi, l'intégration d'un système de gestion de l'énergie du bâtiment à notre outil complètera l'étude.

Une autre perspective est l'amélioration de la méthode d'optimisation proposée. Le réseau de neurones utilisé provoque des erreurs lors de l'estimation de la valeur de fitness. Une étude sur la sélection des exemples d'apprentissage pourrait être une piste de solution envisageable. Aussi, un ré-apprentissage récursif du RNA pourrait minimiser cette erreur. Pour ce qui est de la méthode *K-means*, le travail pourrait être complété par l'étude du nombre de classes à utiliser ou bien, choisir une autre méthode de classification qui ne nécessite pas une connaissance a priori du nombre de classes.

Il existe plusieurs problèmes qui n'ont pas été explorés dans ce travail, comprenant l'étude et la planification de capteurs mobiles, l'étude de l'impact du déplacement des usagers sur les performances du réseau. Aussi, les performances énergétiques du bâtiment n'ont pas été étudiées. Pour cela, d'avantage d'efforts devraient être faits afin d'apporter une contribution à ce domaine.

Implémentation de ce travail

Ce travail fait partie du Work Package 1 du projet Européen CREST partiellement financé par le programme Interreg IVA France-Manche via les fonds FEDER. Le projet regroupe 5 partenaires : SCIC Pays de Dinan, Cambridgeshire County Council, Southend-on-Sea Borough Council, Etudes et Chantiers Bretagne, Pays de Loire et le groupe CESI avec un budget global de 1.8 Millions d'euros. Le projet a été officiellement lancé en Octobre 2011 sur une période de 36 mois [198]. Les objectifs principaux du projet étaient :

- proposer une nouvelle approche entièrement intégrée aux changements d'énergie nécessaires dans les bâtiments existants et leur éco-utilisation dans la Manche avec un accent sur la communauté éducative ;

- Tester des outils ou de nouvelles utilisations par un laboratoire vivant et des services dédiés, et impliquer les utilisateurs à un stade précoce dans la conception de nouveaux produits / usages ;
- élaborer des programmes de formation de courte durée pratiques et interactives sur l'efficacité énergétique, l'énergie durable et bâtiment écologique.

Le groupe de travail WP1 s'est intéressé à :

- Etudier le comportement des usagers dans différents types de bâtiments dédiés à l'enseignement et à la formation (école, université, bâtiment administratif) ;
- Evaluer le comportement des usagers dans le contexte de l'ambiance intelligente. Comportement face à différentes interfaces hommes machines fixe (tablette tactile) ou nomade (smartphone) et permettant d'interagir avec des objets intelligents dans le bâtiment ;
- Etudier l'implantation et la fiabilité des réseaux de capteurs et des données associées.

Cette thèse s'intègre dans le troisième objectif du WP1. L'objectif était de développer une modélisation, simulation et outil de déploiement des RCSFs pour une intégration optimale de la technologie dans le bâtiment. Dans le projet, l'outil devrait disposer d'une interface graphique simple et facile à utiliser, qui comporte un modèle de communication (propagation d'ondes) et prend en compte les contraintes liées à l'environnement de déploiement (bâtiment). Grâce à cet outil, les concepteurs de RCSF seront assistés durant la phase de déploiement et ainsi économiser le temps et les ressources nécessaires.

Publications basées sur ce travail

Cette section présente la liste complète des publication résultant de cette thèse :

Revue scientifiques avec CL

1. BENATIA, M., SAHNOUN, M., LOUIS, A., BAUDRY, D., MAZARI, M., & EL-HAMI, *Multi-Objective WSN Deployment Using Genetic Algorithms under Cost, Coverage, and Connectivity Constraints*. accepté dans la revue : Wireless Personal Networks (Springer).
2. BENATIA, M., SAHNOUN, M., LOUIS, A., BAUDRY, D., MAZARI, M., & EL-HAMI, *A Comparative Study Between Evolutionary Based Methods To Solve WSN Deployment Problem In indoor applications*, Springer book "Recent developments of metaheuristics (invité suite à la conference META'14)

Conférences internationales avec CL

1. BENATIA, M. A., LOUIS, A., BAUDRY, D., MAZARI, B., & EL HAMI, A. (2014, May). *WSN's modeling for a smart building application*. In IEEE International Energy Conference (ENERGYCON'14), (pp. 821-827). IEEE.
2. BENATIA, M. A., LOUIS, A., BAUDRY, D., MAZARI, B., & EL HAMI, A. (2014, June). *Impact of radio propagation in buildings on WSN's lifetime*. In Global Summit on Computer & Information Technology (GSCIT'14), (pp. 1-6). IEEE.
3. BENATIA, M., SAHNOUN, M., LOUIS, A., BAUDRY, D., MAZARI, M., & ELHAMI, *Optimized Sink node Deployment in WSN Using Genetic Algorithms through Coverage and Cost Constraints*. International Conference in Metaheuristics (META'14), 2014

Conférence nationale avec CL

1. BENATIA, M., SAHNOUN, M., LOUIS, A., BAUDRY, D., MAZARI, M., & EL-HAMI, *Méta heuristiques pour le Placement Optimisé des Noeuds Routeurs Dans un Réseau de Capteurs Sans Fil*, ROADEF'15.

Livrable Projet CREST

1. BENATIA, M. A., BAUDRY, D., *Livrable projet CREST* (2014).

Publication en cours de rédaction

1. BENATIA, M., SAHNOUN, M., LOUIS, A., BAUDRY, D., MAZARI, M., & EL-HAMI, *Hybrid Metaheuristic for the WSN Deployment Problem*, Journal à définir.

Bibliographie

- [1] Jaap C Haartsen. The bluetooth radio system. *Personal Communications, IEEE*, 7(1) :28–36, 2000.
- [2] Nick Hunn. An introduction to wibree. *White paper*, 2006.
- [3] Kazimierz Siwiak. *Ultra-Wideband Radio*. Wiley Online Library, 2004.
- [4] Sinem Coleri Ergen. Zigbee/ieee 802.15. 4 summary. *UC Berkeley, September*, 10 : 17, 2004.
- [5] Valérie Deroin. Diffusion et utilisation des tic en france et en europe en 2009. *Culture chiffres*, (2) :1–12, 2010.
- [6] Alejandro Aragon-Zavala. *Antennas and propagation for wireless communication systems*. John Wiley & Sons, 2008.
- [7] Olivier Bouchet, Hervé Sizun, Christian Boisrobert, Frédérique de Fornel, and Pierre-Noël Favennec. Line of sight propagation. *Free-Space Optics : Propagation and Communication*, pages 77–128, 2006.
- [8] Chuan Zhu, Chunlin Zheng, Lei Shu, and Guangjie Han. A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Network and Computer Applications*, 35(2) :619–632, 2012.
- [9] Nadjib Ait Saadi. *Multi-objective wireless sensor network deployment*. PhD thesis, Paris 6, 2010.
- [10] Jiaying Song and Yen Kheng Tan. Energy consumption analysis of zigbee-based energy harvesting wireless sensor networks. In *Communication Systems (ICCS), 2012 IEEE International Conference on*, pages 468–472. IEEE, 2012.
- [11] Vincent Gardeux. *Conception d’heuristiques d’optimisation pour les problèmes de grande dimension. Application à l’analyse de données de puces à ADN*. PhD thesis, Université de Paris-Est, 2012.

- [12] Omessaad HaJJI. *Contribution au développement de méthodes d'optimisation stochastiques. Application à la conception des dispositifs électrotechniques*. PhD thesis, Ecole Centrale de Lille, 2003.
- [13] Mouadh Yagoubi. *Optimisation évolutionnaire multi-objectif parallèle : application à la combustion Diesel*. PhD thesis, Université Paris Sud-Paris XI, 2012.
- [14] Christopher L Holloway, Patrick L Perini, Ronald R DeLyser, and K Allen. Analysis of composite walls and their effects on short-path propagation modeling. *Vehicular Technology, IEEE Transactions on*, 46(3) :730–738, 1997.
- [15] El-Ghazali Talbi. A unified taxonomy of hybrid metaheuristics with mathematical programming, constraint programming and machine learning. In *Hybrid Metaheuristics*, pages 3–76. Springer, 2013.
- [16] David J Livingstone. Artificial neural networks. *Method sand Applications Series : Methods in Molecular Biology*, 458, 2009.
- [17] B Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.
- [18] Jin-Shyan Lee, Yu-Wei Su, and Chung-Chou Shen. A comparative study of wireless protocols : Bluetooth, uwb, zigbee, and wi-fi. In *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pages 46–51. IEEE, 2007.
- [19] Daphne M Simmonds and Anol Bhattacharjee. Smart systems, smarter living : An empirical study of the building automation system in organizations. 2015.
- [20] Chun-Wei Tsai. An effective wsn deployment algorithm via search economics. *Computer Networks*, 101 :178–191, 2016.
- [21] Auda Raheemah, Naseer Sabri, MS Salim, Phaklen Ehkan, R Kamaruddin, R Badlishah Ahmad, MN Jaafar, SA Aljunid, and MH Chemat. Influences of parts of tree on propagation path losses for wsn deployment in greenhouse environments. *Journal of Theoretical and Applied Information Technology*, 81(3) :552, 2015.
- [22] Ahmed Boubrima, Frédéric Matigot, Walid Bechkit, Hervé Rivano, and Anne Ruas. Optimal deployment of wireless sensor networks for air pollution monitoring. In *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7. IEEE, 2015.
- [23] Sabbani Rama Devi and A Aruna Jyothi. Classification of wsn deployment schemes. *IJCER*, 4(4) :78–82, 2015.

- [24] Wenliang Du, Jing Deng, Yungshiang S Han, Shigang Chen, and Pramod K Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1. IEEE, 2004.
- [25] Mustapha Reda Senouci, Mohamed El Yazid Boudaren, Mohamed Abdelkrim Senouci, and Abdelhamid Mellouk. A smart methodology for deterministic deployment of wireless sensor networks. In *Smart Communications in Network Technologies (SaCoNeT), 2014 International Conference on*, pages 1–6. IEEE, 2014.
- [26] Raghavendra V Kulkarni and Ganesh Kumar Venayagamoorthy. Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, 40(6) :663–675, 2010.
- [27] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97. ACM, 2002.
- [28] Wang Yiyue, Liao Hongmei, and Hu Hengyang. Wireless sensor network deployment using an optimized artificial fish swarm algorithm. In *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, volume 2, pages 90–94. IEEE, 2012.
- [29] Soumyadip Sengupta, Swagatam Das, MD Nasir, and Bijaya K Panigrahi. Multi-objective node deployment in wsns : In search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity. *Engineering Applications of Artificial Intelligence*, 26(1) :405–416, 2013.
- [30] Antony Guinard, Alan McGibney, and Dirk Pesch. A wireless sensor network design tool to support building energy management. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 25–30. ACM, 2009.
- [31] Hervé Lefebvre. *La domotique : le rôle des réseaux dans un processus d'innovation interactive*. PhD thesis, Grenoble 2, 1997.
- [32] Jean-Baptiste Waldner and Jean Baptiste Waldner. *Nano-informatique et intelligence ambiante : inventer l'ordinateur du XXIe siècle*. Hermès Science, 2007.
- [33] Hannes Hartenstein and Kenneth P Laberteaux. A tutorial survey on vehicular ad hoc networks. *Communications Magazine, IEEE*, 46(6) :164–171, 2008.

- [34] Sven Meyer and Andry Rakotonirainy. A survey of research on context-aware homes. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21*, pages 159–168. Australian Computer Society, Inc., 2003.
- [35] Xi Fang, Satyajayant Misra, Guoliang Xue, and Dejun Yang. Smart grid—the new and improved power grid : A survey. *Communications Surveys & Tutorials, IEEE*, 14(4) :944–980, 2012.
- [36] Anne Cancellieri. *L’habitat du futur : défis et prospective pour le prochain quart de siècle*. Documentation française, 1992.
- [37] Pierre Brun and Edmond-Antoine Decamps. *La domotique*. Presses universitaires de France, 1992.
- [38] W Keith Edwards and Rebecca E Grinter. At home with ubiquitous computing : seven challenges. In *Ubicomp 2001 : Ubiquitous Computing*, pages 256–272. Springer, 2001.
- [39] Marie Chan, Daniel Estève, Christophe Escriba, and Eric Campo. A review of smart homes—present state and future challenges. *Computer methods and programs in biomedicine*, 91(1) :55–81, 2008.
- [40] TRON Architecture Design by Ken Sakamura. Toyota dream house (papi), 2010. URL <http://tronweb.super-nova.co.jp/toyotadreamhousepapi.html>.
- [41] LAAS-CNRS. Architectures dynamiques reconfigurables pour systèmes embarqués autonomes mobiles, 2010. URL <https://www.laas.fr/public/fr/adream>.
- [42] Alain Anfosso and Stéphane Rebaudo. Gérontechnologies et contrôle de l’environnement au service du maintien à domicile : le projet gerhome. *Gérontologie et société*, 136(1) :119–131, 2011.
- [43] Nadia Zouba, Francois Bremond, and Monique Thonnat. Multisensor fusion for monitoring elderly activities at home. In *Advanced Video and Signal Based Surveillance, 2009. AVSS’09. Sixth IEEE International Conference on*, pages 98–103. IEEE, 2009.
- [44] Nadia Zouba, François Brémond, Monique Thonnat, Alain Anfosso, Eric Pascual, Patrick Mallea, Veronique Mailland, and Olivier Guerin. A computer system to monitor older adults at home : Preliminary results. In *Gerontechnology Journal*, volume 8, pages 129–139, 2009.

- [45] Nadia Zouba, Bernard Boulay, Francois Bremond, and Monique Thonnat. Monitoring activities of daily living (adls) of elderly based on 3d key human postures. In *Cognitive Vision*, pages 37–50. Springer, 2008.
- [46] Mathieu Gallissot, Jean Caelen, Francis Jambon, and Brigitte Meillon. Une plateforme usage pour l’intégration de l’informatique ambiante dans l’habitat : L’appartement domus. *TSI. Technique et science informatiques*, 32(5) :547–574, 2013.
- [47] Michel Vacher. *Analyse sonore et multimodale dans le domaine de l’assistance à domicile*. PhD thesis, Université de Grenoble, 2011.
- [48] Pedro Chahuara Quispe. *Contrôle intelligent de la domotique à partir d’informations temporelles multi sources imprécises et incertaines*. PhD thesis, Université de Grenoble, 2013.
- [49] Michel Vacher, Dan Istrate, François Portet, Thierry Joubert, Thierry Chevalier, Serge Smidtas, Brigitte Meillon, Benjamin Lecouteux, Mohamed Sehili, Pedro Chahuara, et al. The sweet-home project : Audio technology in smart homes to improve well-being and reliance. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 5291–5294. IEEE, 2011.
- [50] Vidyasagar Potdar, Atif Sharif, and Elizabeth Chang. Wireless sensor networks : A survey. In *Advanced Information Networking and Applications Workshops, 2009. WAINA ’09. International Conference on*, pages 636–641. IEEE, 2009.
- [51] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12) :2292–2330, 2008.
- [52] Milica Pejanovic Durisic, Zhilbert Tafa, Goran Dimic, and Veljko Milutinovic. A survey of military applications of wireless sensor networks. In *Embedded Computing (MECO), 2012 Mediterranean Conference on*, pages 196–199. IEEE, 2012.
- [53] Th Arampatzis, John Lygeros, and S Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, pages 719–724. IEEE, 2005.
- [54] Alan Lelah, Fabrice Mathieux, and Daniel Brissaud. Contributions to eco-design of machine-to-machine product service systems : the example of waste glass collection. *Journal of Cleaner Production*, 19(9) :1033–1044, 2011.

- [55] Byungrak Son, Yong-sork Her, and Jung-Gyu Kim. A design and implementation of forest-fires surveillance system based on wireless sensor networks for south korea mountains. *International Journal of Computer Science and Network Security (IJCSNS)*, 6(9) :124–130, 2006.
- [56] Christophe Jacquet, Yacine Bellik, René Farcy, and Yolaine Bourda. Aides électroniques pour le déplacement des personnes non-voyantes : vue d’ensemble et perspectives. In *Proceedings of the 16th conference on Association Francophone d’Interaction Homme-Machine*, pages 93–100. ACM, 2004.
- [57] John A Stankovic. Wireless sensor networks. *IEEE Computer*, 41(10) :92–95, 2008.
- [58] Ian F Akyildiz and Mehmet Can Vuran. *Wireless sensor networks*, volume 4. John Wiley & Sons, 2010.
- [59] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks : a survey. *Computer networks*, 38(4) :393–422, 2002.
- [60] Shoa-Yei Yeong, Tat-Chee Wan, and Wafaa Al-Salihiy. Combination of neural network based clustering and genetic algorithm for multi-objective 802.11 n planning. In *Communications (MICC), 2009 IEEE 9th Malaysia International Conference on*, pages 852–856. IEEE, 2009.
- [61] Motorola. Lanplanner, 2015. URL https://portal.motorolasolutions.com/Support/US-EN/Wireless+Networks/WLAN+Management/Planning+&+Individual+Solutions/LANPlanner_US-EN.
- [62] CISCO. Cisco prime infrastructure, 2015. URL <http://www.cisco.com/c/en/us/products/cloud-systems-management/prime-infrastructure/index.html>.
- [63] Alan Mc Gibney, Martin Klepal, and Dirk Pesch. A wireless local area network modeling tool for scalable indoor access point placement optimization. In *Proceedings of the 2010 Spring Simulation Multiconference*, page 163. Society for Computer Simulation International, 2010.
- [64] Jan Beutel, Kay Römer, Matthias Ringwald, and Matthias Woehrle. Deployment techniques for sensor networks. In *Sensor Networks*, pages 219–248. Springer, 2009.
- [65] Juan A Gomez-Pulido and Jose M Lanza-Gutierrez. Reliability and efficiency in wireless sensor networks : heuristic approaches. *Journal of Heuristics*, pages 1–3.
- [66] James Clerk Maxwell. *A treatise on electricity and magnetism*, volume 1. Clarendon press, 1881.

- [67] John Kraus. Antennas since hertz and marconi. *IEEE Transactions on Antennas and Propagation*, pages 131–137, 1985.
- [68] CA Balannis. Antenna theory : Analisis and desain, 1982.
- [69] GE Athanaasiadou, AR Nix, and JP McGeehan. A new 3d indoor ray-tracing propagation model with particular reference to the prediction of power and delay spread. In *Personal, Indoor and Mobile Radio Communications, 1995. PIMRC'95. Wireless : Merging onto the Information Superhighway., Sixth IEEE International Symposium on*, volume 3, page 1161. IEEE, 1995.
- [70] Chang-Fa Yang, Boau-Cheng Wu, and Chuen-Jyi Ko. A ray-tracing method for modeling indoor wave propagation and penetration. *Antennas and Propagation, IEEE Transactions on*, 46(6) :907–919, 1998.
- [71] Zhong Ji, Bin-Hong Li, Hao-Xing Wang, Hsing-Yi Chen, and Yaw-Gen Zhau. A new indoor ray-tracing propagation prediction model. In *Computational Electromagnetics and Its Applications, 1999. Proceedings.(ICCEA'99) 1999 International Conference on*, pages 540–542. IEEE, 1999.
- [72] Abdullah E Shaikh, Faisal Majeed, Muhammad Zeeshan, Tahir Rabbani, and Imran Sheikh. Efficient implementation of deterministic 3-d ray tracing model to predict propagation losses in indoor environments. In *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, volume 3, pages 1208–1212. IEEE, 2002.
- [73] Matthias Lott and Ingo Forkel. A multi-wall-and-floor model for indoor radio propagation. In *Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd*, volume 1, pages 464–468. IEEE, 2001.
- [74] B Sujak, D Kumar Ghodgaonkar, BM Ali, and S Khatun. Indoor propagation channel models for wlan 802.11 b at 2.4 ghz ism band. In *Applied Electromagnetics, 2005. APACE 2005. Asia-Pacific Conference on*, pages 5–pp. IEEE, 2005.
- [75] Theofilos Chrysikos, Giannis Georgopoulos, and Stavros Kotsopoulos. Wireless channel characterization for a home indoor propagation topology at 2.4 ghz. In *Wireless Telecommunications Symposium (WTS), 2011*, pages 1–10. IEEE, 2011.
- [76] Xi Shen, Kun Xu, Xiaoqiang Sun, Jian Wu, and Jintong Lin. Optimized indoor wireless propagation model in wifi-rof network architecture for rss-based localization in the internet of things. In *Microwave Photonics, 2011 International Topical Meeting on & Microwave Photonics Conference, 2011 Asia-Pacific, MWP/APMP*, pages 274–277. IEEE, 2011.

- [77] Constantine A Balanis. *Antenna theory : analysis and design*. John Wiley & Sons, 2012.
- [78] Abdollah Ghasemi, Ali Abedi, and Farshid Ghasemi. Line of sight propagation. In *Propagation Engineering in Wireless Communications*, pages 289–355. Springer, 2012.
- [79] JM Keenan and AJ Motley. Radio coverage in buildings. *British telecom technology Journal*, 8(1) :19–24, 1990.
- [80] Ashraf Hossain, PK Biswas, and S Chakrabarti. Sensing models and its impact on network coverage in wireless sensor network. In *2008 IEEE Region 10 and the Third international Conference on Industrial and Information Systems*, pages 1–5. IEEE, 2008.
- [81] Celal Öztürk, Derviş Karaboğa, and BEYZA GÖRKEMLİ. Artificial bee colony algorithm for dynamic deployment of wireless sensor networks. *Turkish Journal of Electrical Engineering & Computer Sciences*, 20(2) :255–262, 2012.
- [82] Nadjib Aitsaadi, Nadjib Achir, Khaled Boussetta, and Guy Pujolle. Artificial potential field approach in wsn deployment : Cost, qom, connectivity, and lifetime constraints. *Computer Networks*, 55(1) :84–105, 2011.
- [83] Wooyoung Kim, Kirill Mechitov, J-Y Choi, and Soo Ham. On target tracking with binary proximity sensors. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 301–308. IEEE, 2005.
- [84] Yao Zou and Krishnendu Chakrabarty. Sensor deployment and target localization based on virtual forces. In *IEEE INFOCOM*, volume 2, pages 1293–1303. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 2003.
- [85] Santpal Singh Dhillon and Krishnendu Chakrabarty. A fault-tolerant approach to sensor deployment in distributed sensor networks. *Dept. of EE & CE, Duke University, Durham, NC*, 2002.
- [86] Santpal S Dhillon, Krishnendu Chakrabarty, and SS Iyengar. Sensor placement for grid coverage under imprecise detections. In *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, volume 2, pages 1581–1587. IEEE, 2002.
- [87] Santpal Singh Dhillon and Krishnendu Chakrabarty. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, volume 3, pages 1609–1614. IEEE, 2003.

- [88] Danping He, Gabriel Mujica, Jorge Portilla, and Teresa Riesgo. Modelling and planning reliable wireless sensor networks based on multi-objective optimization genetic algorithm with changeable length. *Journal of Heuristics*, 21(2) :257–300, 2015.
- [89] Danping He, Jorge Portilla, and Teresa Riesgo. A 3d multi-objective optimization planning algorithm for wireless sensor networks. In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, pages 5428–5433. IEEE, 2013.
- [90] Nadjib Aitsaadi, Nadjib Achir, Khaled Boussetta, and Guy Pujolle. A tabu search wsn deployment method for monitoring geographically irregular distributed events. *Sensors*, 9(3) :1625–1643, 2009.
- [91] Kenan Xu, Quanhong Wang, Hossam Hassanein, and Glen Takahara. Optimal wireless sensor networks (wsns) deployment : minimum cost with lifetime constraint. In *WiMob'2005), IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005.*, volume 3, pages 454–461. IEEE, 2005.
- [92] Fadi M Al-Turjman, Ashraf E Al-Fagih, and Hossam S Hassanein. A novel cost-effective architecture and deployment strategy for integrated rfid and wsn systems. In *Computing, Networking and Communications (ICNC), 2012 International Conference on*, pages 835–839. IEEE, 2012.
- [93] Yunxia Chen and Qing Zhao. On the lifetime of wireless sensor networks. *Communications Letters, IEEE*, 9(11) :976–978, 2005.
- [94] Matthieu Le Berre. *Optimisation de déploiement et localisation de cible dans les réseaux de capteurs*. PhD thesis, Troyes, 2014.
- [95] Gregory J Pottie and William J Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5) :51–58, 2000.
- [96] G Dimitriou, PK Kikiras, Georgios I Stamoulis, and IN Avaritsiotis. A tool for calculating energy consumption in wireless sensor networks. In *Advances in Informatics*, pages 611–621. Springer, 2005.
- [97] José Manuel Lanza-Gutiérrez, Juan Antonio Gómez-Pulido, Miguel A Vega-Rodríguez, and Juan Manuel Sánchez-Pérez. Optimizing energy consumption in heterogeneous wireless sensor networks by means of evolutionary algorithms. In *Applications of Evolutionary Computation*, pages 1–10. Springer, 2012.
- [98] Agustin Barberis, Leonardo Barboni, and Maurizio Valle. Evaluating energy consumption in wireless sensor networks applications. In *Digital System Design*

- Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on*, pages 455–462. IEEE, 2007.
- [99] Jia Yunjie, Liu Ming, Zhu Song, and Dong Pengtao. A clustering routing algorithm based on energy and distance in wsn. In *Computer Distributed Control and Intelligent Environmental Monitoring (CDCIEM), 2012 International Conference on*, pages 9–12. IEEE, 2012.
- [100] Zayneb Trabelsi Ayoub, Sofiane Ouni, and Farouk Kamoun. Energy consumption analysis to predict the lifetime of ieee 802.15. 4 wireless sensor networks. In *Communications and Networking (ComNet), 2012 Third International Conference on*, pages 1–6. IEEE, 2012.
- [101] Qin Wang and Woodward Yang. Energy consumption model for power management in wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on*, pages 142–151. IEEE, 2007.
- [102] Hui Wang, H Eduardo Roman, Liyong Yuan, Yongfeng Huang, and Rongli Wang. Connectivity, coverage and power consumption in large-scale wireless sensor networks. *Computer Networks*, 75 :212–225, 2014.
- [103] George Bravos and Athanassios G Kanatas. Energy consumption and trade-offs on wireless sensor networks. In *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, volume 2, pages 1279–1283. IEEE, 2005.
- [104] Ruihua Zhang, Dongfeng Yuan, and Quanquan Liang. Analysis of energy consumption and lifetime of wireless sensor networks. In *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*, pages 1159–1164. IEEE, 2007.
- [105] Zongheng Zhou, Samir Das, and Himanshu Gupta. Connected k-coverage problem in sensor networks. In *Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on*, pages 373–378. IEEE, 2004.
- [106] Nitin Mittal, Davinder Pal Singh, Amanjeet Panghal, and RS Chauhan. Improved leach communication protocol for wsn. In *National Conference on Computational Instrumentation*, volume 3, pages 153–156, 2010.
- [107] Zbigniew Michalewicz and David B Fogel. *How to solve it : modern heuristics*. Springer Science & Business Media, 2013.

- [108] Chi-Fu Huang and Yu-Chee Tseng. A survey of solutions for the coverage problems in wireless sensor networks. *Wireless Personal Communications*, 6(1) :1–8, 2005.
- [109] Anju Sangwan and Rishi Pal Singh. Survey on coverage problems in wireless sensor networks. *Wireless Personal Communications*, 80(4) :1475–1500, 2015.
- [110] Hua Su, Gaoyong Wang, Xuemei Sun, and Dong Yu. Optimal node deployment strategy for wireless sensor networks based on dynamic ant colony algorithm. *International Journal of Embedded Systems*, 8(2-3) :258–265, 2016.
- [111] D Lee and Arthurk Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2) :276–282, 1986.
- [112] Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. *Mobile Networks and Applications*, 10(4) :519–528, 2005.
- [113] Héctor González-Baños. A randomized art-gallery algorithm for sensor placement. In *Proceedings of the seventeenth annual symposium on Computational geometry*, pages 232–240. ACM, 2001.
- [114] Mohamed Younis and Kemal Akkaya. Strategies and techniques for node placement in wireless sensor networks : A survey. *Ad Hoc Networks*, 6(4) :621–655, 2008.
- [115] Jose M Lanza-Gutierrez, Juan A Gomez-Pulido, Silvio Priem-Mendes, M Ferreira, and JS Pereira. Planning the deployment of indoor wireless sensor networks through multiobjective evolutionary techniques. In *European Conference on the Applications of Evolutionary Computation*, pages 128–139. Springer, 2015.
- [116] Khash-Erdene Jalsan, Rohan N Soman, Kallirroi Flouri, Marios A Kyriakides, Glauco Feltrin, and Toula Onoufriou. Layout optimization of wireless sensor networks for structural health monitoring. *Smart Structures and Systems*, 14(1) : 39–54, 2014.
- [117] Xinyi Tan, Fuquan Zhang, Kaiqi Chen, and Demin Gao. A survey on deployment of wireless sensor network. 2016.
- [118] BS Panda and D Pushparaj Shetty. Strong minimum energy hierarchical topology in wireless sensor networks. *Journal of Combinatorial Optimization*, pages 1–14, 2015.
- [119] Jae-Hwan Chang and Leandros Tassioulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 12(4) :609–619, 2004.

- [120] Wei-Lun Chang, Deze Zeng, Rung-Ching Chen, and Song Guo. An artificial bee colony algorithm for data collection path planning in sparse wireless sensor networks. *International Journal of Machine Learning and Cybernetics*, 6(3) :375–383, 2015.
- [121] A Attea Bara’a, Enan A Khalil, Suat Özdemir, and Oktay Yıldız. A multi-objective disjoint set covers for reliable lifetime maximization of wireless sensor networks. *Wireless Personal Communications*, 81(2) :819–838, 2015.
- [122] Omer Ozkan, Murat Ermis, and Ilker Bekmezci. A hybrid matheuristic approach for designing reliable wireless multimedia sensor networks. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 875–882. ACM, 2015.
- [123] Yves Cherruault. *Optimisation : méthodes locales et globales*. Presses Universitaires de France-PUF, 1999.
- [124] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.
- [125] Ding-Zhu Du and Panos M Pardalos. *Handbook of combinatorial optimization : supplement*, volume 1. Springer Science & Business Media, 2013.
- [126] David E Goldberg. *The design of innovation : Lessons from and for competent genetic algorithms*, volume 7. Springer Science & Business Media, 2013.
- [127] Alain Berro. *Optimisation multiobjectifs et stratégies d’évolution en environnement dynamique*. PhD thesis, ANRT [diff.], 2001.
- [128] Gérard Berthiau and Patrick Siarry. État de l’art des méthodes “d’optimisation globale”. *RAIRO-Operations Research*, 35(3) :329–365, 2001.
- [129] Jin-Kao Hao, Philippe Galinier, and Michel Habib. Métaheuristiques pour l’optimisation combinatoire et l’affectation sous contraintes. *Revue d’intelligence artificielle*, 13(2) :283–324, 1999.
- [130] Eugene L Lawler and David E Wood. Branch-and-bound methods : A survey. *Operations research*, 14(4) :699–719, 1966.
- [131] Raksmeay Phan. *Méthodes exactes et approchées par partition en cliques de graphes*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2013.
- [132] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4) :308–313, 1965.

-
- [133] Michel Minoux. *Programmation mathématique : théorie et algorithmes*, volume 1. Dunod Paris, 1983.
- [134] Tobias Achterberg. Scip : solving constraint integer programs. *Mathematical Programming Computation*, 1(1) :1–41, 2009.
- [135] Robert A Stubbs and Sanjay Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical programming*, 86(3) :515–532, 1999.
- [136] James C Spall. *Introduction to stochastic search and optimization : estimation, simulation, and control*, volume 65. John Wiley & Sons, 2005.
- [137] Fred Glover and Gary A Kochenberger. *Handbook of metaheuristics*. Springer Science & Business Media, 2003.
- [138] El-Ghazali Talbi. *Metaheuristics : from design to implementation*, volume 74. John Wiley & Sons, 2009.
- [139] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization : Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3) : 268–308, 2003.
- [140] Jason Brownlee. *Clever algorithms : nature-inspired programming recipes*. Jason Brownlee, 2011.
- [141] Yezid Donoso and Ramon Fabregat. *Multi-objective optimization in computer networks using metaheuristics*. CRC Press, 2016.
- [142] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5) :533–549, 1986.
- [143] Charles Darwin. Evolution. *Natural Selection*, 1807.
- [144] Lawrence J Fogel, Alvin J Owens, and Michael J Walsh. Artificial intelligence through simulated evolution. 1966.
- [145] John H Holland. Adaptation in natural and artificial system : an introduction with application to biology, control and artificial intelligence. *Ann Arbor, University of Michigan Press*, 1975.
- [146] David E Goldberg, Bradley Korb, and Kalyanmoy Deb. Messy genetic algorithms : Motivation, analysis, and first results. *Complex systems*, 3(5) :493–530, 1989.
- [147] I Rechenberg. Evolution strategie : Optimierung technischer systemenach prinzipiender biologischen evolution. *Friedrich Fromannerlag, Stuttgart*, 1(9) :7.

- [148] John R Koza. *Genetic programming : on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.
- [149] John R Koza. *Genetic programming II : automatic discovery of reusable programs*. MIT press, 1994.
- [150] CRYPTOGRAPHIE ET SÉCURITÉ DE. Cours des méthodes de résolution exactes heuristiques et métaheuristiques.
- [151] Russell C Eberhart and Yuhui Shi. Particle swarm optimization : developments, applications and resources. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 1, pages 81–86. IEEE, 2001.
- [152] Antoine Dutot and Damien Olivier. Optimisation par essaim de particules application au problème des n-reines. *Laboratoire Informatique du Havre, Université du Havre*, 2002.
- [153] Yann Cooren. *Perfectionnement d'un algorithme adaptatif d'Optimisation par Essaim Particulaire : application en génie médical et en électronique*. PhD thesis, Université Paris-Est, 2008.
- [154] Guillaume CALAS. Optimisation par essaim particulaire. *Une*, 3 :3.
- [155] Eckart Zitzler. Evolutionary algorithms for multiobjective optimization : Methods and applications. 1999.
- [156] Joon-Yong Lee, Joon-Hong Seok, and Ju-Jang Lee. Multiobjective optimization approach for sensor arrangement in a complex indoor environment. *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, 42 (2) :174–186, 2012.
- [157] Konstantinos P Ferentinos and Theodore A Tsiligiridis. Adaptive design optimization of wireless sensor networks using genetic algorithms. *Computer Networks*, 51(4) :1031–1051, 2007.
- [158] Andreas Konstantinidis, Kun Yang, Qingfu Zhang, and Demetrios Zeinalipour-Yazti. A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks. *Computer networks*, 54(6) :960–976, 2010.
- [159] Yourim Yoon and Yong-Hyuk Kim. An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks. *Cybernetics, IEEE Transactions on*, 43(5) :1473–1483, 2013.

- [160] Xiao-Min Hu, Jun Zhang, Yan Yu, Henry Shu-Hung Chung, Yuan-Long Li, Yu-Hui Shi, and Xiao-Nan Luo. Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks. *Evolutionary Computation, IEEE Transactions on*, 14(5) :766–781, 2010.
- [161] Alfredo J Perez, Miguel Labrador, Pedro M Wightman, et al. A multiobjective approach to the relay placement problem in wsns. In *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pages 475–480. IEEE, 2011.
- [162] Evren Güney, Necati Aras, İ Kuban Altınel, and Cem Ersoy. Efficient integer programming formulations for optimum sink location and routing in heterogeneous wireless sensor networks. *Computer Networks*, 54(11) :1805–1822, 2010.
- [163] Alessandro Pinto, Massimiliano D’Angelo, Carlo Fischione, Eelco Scholte, and Alberto Sangiovanni-Vincentelli. Synthesis of embedded networks for building automation and control. In *American Control Conference, 2008*, pages 920–925. IEEE, 2008.
- [164] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm : Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2) :182–197, 2002.
- [165] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms : Empirical results. *Evolutionary computation*, 8(2) :173–195, 2000.
- [166] David W Corne, Joshua D Knowles, and Martin J Oates. The pareto envelope-based selection algorithm for multiobjective optimization. In *Parallel Problem Solving from Nature PPSN VI*, pages 839–848. Springer, 2000.
- [167] Rainer Storn and Kenneth Price. *Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces*, volume 3. ICSI Berkeley, 1995.
- [168] Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4) :341–359, 1997.
- [169] Yann Collette and Patrick Siarry. *Multiobjective optimization : principles and case studies*. Springer Science & Business Media, 2013.
- [170] Eckart Zitzler, Marco Laumanns, Lothar Thiele, Eckart Zitzler, Eckart Zitzler, Lothar Thiele, and Lothar Thiele. *Spea2 : Improving the strength pareto evolutionary algorithm*, 2001.

- [171] Gary Pampara, Andries Petrus Engelbrecht, and Nelis Franken. Binary differential evolution. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1873–1879. IEEE, 2006.
- [172] Yunjiao Xue, Ho Sung Lee, Ming Yang, Priyantha Kumarawadu, Hamada H Ghenniwa, and Weiming Shen. Performance evaluation of ns-2 simulator for wireless sensor networks. In *2007 Canadian Conference on Electrical and Computer Engineering*, pages 1372–1375. IEEE, 2007.
- [173] Charlie Vanaret. Hybridation d’algorithmes évolutionnaires et de méthodes d’intervalles pour l’optimisation de problèmes difficiles. 2015.
- [174] Christian Blum, Jakob Puchinger, Günther R Raidl, and Andrea Roli. Hybrid metaheuristics in combinatorial optimization : A survey. *Applied Soft Computing*, 11(6) :4135–4151, 2011.
- [175] El-Ghazali Talbi. *Hybrid metaheuristics*, volume 434. Springer, 2013.
- [176] Una Benlic and Jin-Kao Hao. Hybrid metaheuristics for the graph partitioning problem. In *Hybrid Metaheuristics*, pages 157–185. Springer, 2013.
- [177] L Shi and K Rasheed. A survey of fitness approximation methods applied in evolutionary algorithms. In *Computational intelligence in expensive optimization problems*, pages 3–28. Springer, 2010.
- [178] Yaochu Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft computing*, 9(1) :3–12, 2005.
- [179] Margarita Reyes-Sierra and Carlos A Coello Coello. A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 65–72. IEEE, 2005.
- [180] António Gaspar-Cunha and Armando Vieira. A multi-objective evolutionary algorithm using neural networks to approximate fitness evaluations. *Int. J. Comput. Syst. Signal*, 6(1) :18–36, 2005.
- [181] Pawan KS Nain and Kalyanmoy Deb. A computationally effective multi-objective search and optimization technique using coarse-to-fine grain modeling. In *IN 2002 PPSN WORKSHOP ON EVOLUTIONARY MULTIOBJECTIVE OPTIMIZAA COMPREHENSIVE SURVEY OF FITNESS APPROXIMATION IN EVOLUTIONARY COMPUTATION 13 TION*. Citeseer, 2002.

- [182] Lars Willmes, Thomas Back, Yaochu Jin, and Bernhard Sendhoff. Comparing neural networks and kriging for fitness approximation in evolutionary optimization. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 1, pages 663–670. IEEE, 2003.
- [183] Yaochu Jin and Bernhard Sendhoff. Reducing fitness evaluations using clustering techniques and neural network ensembles. In *Genetic and Evolutionary Computation Conference*, pages 688–699. Springer, 2004.
- [184] Robert G Beiko and Robert L Charlebois. Gann : genetic algorithm neural networks for the detection of conserved combinations of features in dna. *BMC bioinformatics*, 6(1) :36, 2005.
- [185] Michael Schmitt. On the complexity of computing and learning with multiplicative neural networks. *Neural Computation*, 14(2) :241–301, 2002.
- [186] Francois E Cellier and Jurgen Greifeneder. *Continuous system modeling*. Springer Science & Business Media, 2013.
- [187] Duc Pham and Dervis Karaboga. *Intelligent optimisation techniques : genetic algorithms, tabu search, simulated annealing and neural networks*. Springer Science & Business Media, 2012.
- [188] Narendran Rajagopalan, C Mala, M Sridevi, and R Hari Prasath. Optimized channel allocation using genetic algorithm and artificial neural networks. In *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011*, pages 645–655. Springer, 2012.
- [189] Bernhard Grill, Andreas Ekelhart, Elmar Kiesling, Christian Stummer, and Christine Strauss. Multi-objective evolutionary optimization of computation-intensive simulations—the case of security control selection. 2015.
- [190] Jong-Won Yoon and Sung-Bae Cho. Fitness approximation for genetic algorithm using combination of approximation model and fuzzy clustering technique. In *IEEE Congress on Evolutionary Computation*, pages 1–6. IEEE, 2010.
- [191] Anil K Jain. Data clustering : 50 years beyond k-means. *Pattern recognition letters*, 31(8) :651–666, 2010.
- [192] Petros Drineas, Alan Frieze, Ravi Kannan, Santosh Vempala, and V Vinay. Clustering large graphs via the singular value decomposition. *Machine learning*, 56 (1-3) :9–33, 2004.

-
- [193] ARIF Muhammad. *Fusion de Données : Ultime Etape de Reconnaissance de Formes, Applications à l'Identification et à l'Authentification*. PhD thesis, Thèse pour obtenir le grade de Docteur de l'université de Tours Soutenue Le 25 novembre, 2005.
- [194] Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. *Neural network design*, volume 20. PWS publishing company Boston, 1996.
- [195] Jiwen Dong, Lei Zhao, and Liang Zhang. Face recognition based on neural network ensemble and feature fusion. In *2013 IEEE Third International Conference on Information Science and Technology (ICIST)*, pages 59–62. IEEE, 2013.
- [196] Lucian Saletè Buriol, Gereon Frahling, Stefano Leonardi, Alberto Marchetti-Spaccamela, and Christian Sohler. Computing clustering coefficients in data streams. In *European Conference on Complex Systems (ECCS)*, 2006.
- [197] Matthias R Brust, Damla Turgut, Carlos HC Ribeiro, and Marcus Kaiser. Is the clustering coefficient a measure for fault tolerance in wireless sensor networks? In *2012 IEEE International Conference on Communications (ICC)*, pages 183–187. IEEE, 2012.
- [198] MA Benatia, D Baudry, and A Louis. *Livrable crest : état de l'art sur les infrastructures réseau du bâtiment intelligent*. 2015.