



**HAL**  
open science

# La résolution du déploiement 3D d'objets connectés sans fil à l'intérieur en utilisant un schéma hybride entre les méthodes géométriques de déploiement et les algorithmes d'optimisation distribués

Wajih Abdallah

## ► To cite this version:

Wajih Abdallah. La résolution du déploiement 3D d'objets connectés sans fil à l'intérieur en utilisant un schéma hybride entre les méthodes géométriques de déploiement et les algorithmes d'optimisation distribués. Algorithme et structure de données [cs.DS]. Université Toulouse le Mirail - Toulouse II, 2022. Français. NNT : 2022TOU20043 . tel-03978621

**HAL Id: tel-03978621**

**<https://theses.hal.science/tel-03978621v1>**

Submitted on 8 Feb 2023

**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.



# THÈSE

En vue de l'obtention du

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 2 - Jean Jaurès

Présentée et soutenue par

**Wajih ABDALLAH**

Le mercredi 13 Juillet 2022

**La résolution du déploiement d'objets connectés sans fil à l'intérieur en utilisant un schéma hybride entre les méthodes géométriques de déploiement et les algorithmes d'optimisation distribués**

Ecole doctorale : **EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité : **Informatique**

Unité de recherche :

**IRIT : Institut de Recherche en Informatique de Toulouse**

**Directeur de Thèse**

Thierry VAL, Professeur à l'Université de Toulouse

### **Jury**

Pr. Ahmad HASSANAT, Université de Mutah Jordan, Rapporteur

Pr. Nawres KHLIFA, Université de Tunis El Manar, Rapporteuse

Dr. Sami MNASRI, Université de Tabuk Arabie Saoudite, Co-Directeur de thèse

Pr. Abdennaceur KACHOURI, Professeur Emérite, Université de Sfax Tunisie, Président

Dr. Réjane DALCE, ISIS Castres France, Examinatrice

# Université Toulouse 2-Jean Jaurès

## Laboratoire IRIT

### THÈSE

Pour obtenir le grade de  
DOCTEUR DE L'UNIVERSITÉ TOULOUSE 2 JEAN JAURÈS  
**Spécialité : Informatique et Télécommunications**

**La résolution du déploiement d'objets connectés sans fil à  
l'intérieur en utilisant un schéma hybride entre les méthodes  
géométriques de déploiement et les algorithmes d'optimisation  
distribués.**

**Wajih ABDALLAH**

Présentée et soutenue publiquement  
Le 13 juillet 2022

#### **Directeur de Thèse**

Thierry VAL, Professeur à l'Université de Toulouse

#### **JURY**

Pr. Ahmad HASSANAT, Université de Mutah Jordan, Rapporteur  
Pr. Nawres KHLIFA, Université de Tunis El Manar, Rapporteur  
Dr. Sami MNASRI, Université de Tabuk Arabie Saoudite, Co-Directeur de thèse  
Pr. Abdennaceur KACHOURI, Professeur Emérite, Université de Sfax Tunisie, Président  
Dr. Réjane DALCE, ISIS Castres France, Examinatrice  
Dr. Laurent ANDRIEUX, Université de Toulouse, invité

## ***Dédicace***

*« A mes parents, mes frères et ma sœur... »*

---

## Remerciements

---

Je remercie tout d'abord le Pr. Thierry VAL, mon directeur de thèse pour son soutien et sa contribution dans cette recherche ainsi que pour sa disponibilité. En particulier, je le remercie pour l'excellence de son accompagnement ainsi que pour la confiance et l'autonomie qu'il m'a accordées. Je lui suis reconnaissant pour ses encouragements et ses conseils qui ont été précieux, sans lesquels ce travail n'aurait pas été possible. J'exprime également ma gratitude à Dr. Sami MNASRI, maître de conférences et co-cadreur de ma thèse. Je le remercie vivement pour ses conseils pertinents, ses commentaires qui m'ont permis d'améliorer le fond et la forme ainsi que pour sa présence et son soutien moral dans les moments de stress et de doutes tout au long de la thèse.

Je remercie chaleureusement le Pr. Ahmad HASSANAT et le Pr. Nawres KHLIFA pour l'intérêt qu'ils ont témoigné à mes travaux en acceptant de les rapporter et de participer au jury de soutenance. Je remercie également le Pr. Abdennaceur KACHOURI, Dr. Réjane DALCE et Dr. Laurent ANDRIEUX pour leur gentillesse et pour avoir accepté d'examiner mon travail de thèse.

Mes remerciements vont également à tous les enseignants, personnels de l'IUT de Blagnac pour la bonne ambiance qu'ils ont créée dans les locaux de l'IUT. Enfin, je remercie tous mes collègues qui ont contribué à ma formation et à tous ceux qui m'ont aidé durant ce travail.

*À vous tous merci !*

# TABLE DES MATIÈRES

<b>LISTE DES FIGURES.....</b>	<b>V</b>
<b>LISTE DES TABLEAUX .....</b>	<b>VIII</b>
<b>LISTE DES ALGORITHMES .....</b>	<b>X</b>
<b>LISTE DES ACRONYMES .....</b>	<b>XI</b>
<b>INTRODUCTION GENERALE .....</b>	<b>1</b>
<b>PARTIE 1 : ÉTAT DE L'ART .....</b>	<b>7</b>
<b>1. LES RESEAUX SANS FIL DE COLLECTE DE DONNEES DE L'IOT: PROBLEMATIQUE ET ETAT DE L'ART .....</b>	<b>8</b>
<b>1.1. INTRODUCTION.....</b>	<b>9</b>
<b>1.2. NOTIONS PRELIMINAIRES SUR LE RESEAU DE COLLECTE DE DONNEES .....</b>	<b>10</b>
1.2.1. Capteurs .....	10
1.2.2. Capteurs mobiles .....	11
1.2.3. Lien de communication.....	11
1.2.4. RCD hétérogène .....	12
1.2.5. RCD homogène .....	12
1.2.6. Nœuds relais .....	12
1.2.7. Nœuds redondants .....	12
1.2.8. Nœuds mobiles .....	12
1.2.9. Puits mobiles .....	12
1.2.10. Notion de trou d'énergie.....	13
<b>1.3. ARCHITECTURE D'UN NŒUD CAPTEUR .....</b>	<b>13</b>
1.3.1. Unité de détection.....	13
1.3.2. Unité de communication .....	14
1.3.3. Unité de traitement .....	14
1.3.4. Unité de stockage.....	14
<b>1.4. CLASSEMENT DE RESEAU SANS FIL DE COLLECTE DES DONNEES.....</b>	<b>15</b>
1.4.1. Les RCD Terrestres .....	16
1.4.2. Les RCD Souterrains .....	16
1.4.3. Les RCD sous-marins .....	16
1.4.4. Les RCD mobiles.....	17
1.4.5. Les RCD multimédias.....	17
<b>1.5. EXIGENCES D'UN RESEAU DE COLLECTE DES DONNEES SANS FILS.....</b>	<b>17</b>
1.5.1. Tolérance aux pannes.....	17

1.5.2. Évolutif.....	17
1.5.3. Longue durée de vie .....	17
1.5.4. Programmable.....	18
1.5.5. Sécurité.....	18
1.5.6. Abordable .....	18
<b>1.6. LES TECHNOLOGIES UTILISEES DANS LES RESEAUX DE COLLECTE DES DONNEES SANS FIL</b> .....	<b>18</b>
1.6.1. LoRa.....	18
1.6.2. SigFox .....	20
1.6.3. NB -IoT .....	21
1.6.4. IEEE 802.15.4.....	21
1.6.5. BLE .....	22
1.6.6. EnOcean .....	23
1.6.7. UWB.....	24
1.6.8. WiFi.....	25
<b>1.7. CONCLUSION.....</b>	<b>27</b>
<b>2. LE PROBLEME DE DEPLOIEMENT OPTIMISE POUR LES RESEAUX DE COLLECTE DE DONNEES</b> .....	<b>10</b>
<b>2.1. INTRODUCTION .....</b>	<b>29</b>
<b>2.2. MODELES DE COUVERTURE .....</b>	<b>29</b>
2.2.1. Modèle de couverture binaire .....	30
2.2.2. Modèles de détection probabilistes .....	31
<b>2.3. COUVERTURE ET CONNECTIVITE.....</b>	<b>32</b>
2.3.1. Couverture statique .....	33
2.3.2. Couverture dynamique .....	33
<b>2.4. APPROCHES DE DEPLOIEMENT DANS LES RCD.....</b>	<b>34</b>
2.4.1. Déploiement aléatoire .....	34
2.4.2. Déploiement déterministe.....	35
<b>2.5. TECHNIQUE DE DEPLOIEMENT CLASSIQUE DANS LES RCD .....</b>	<b>35</b>
2.5.1. Technique basée sur la force.....	36
2.5.2. Technique basée sur la grille .....	38
2.5.3. Techniques basées sur la géométrie computationnelle .....	41
<b>2.6. LES APPROCHES D'OPTIMISATION DANS LE DEPLOIEMENT DE RESEAUX SANS FIL DE COLLECTE DES DONNEES</b> .....	<b>43</b>
2.6.1. Concept général d'optimisation et ses catégories.....	43
2.6.2. Méthodes de déploiement dans RCD basées sur la méta-heuristique.....	45
2.6.3. Hybridation entre DV et des algorithmes d'optimisation .....	52

<b>2.7. TYPES D'ARCHITECTURE PARALLELES ET DISTRIBUEES POUR LES ALGORITHMES     D'OPTIMISATION.....</b>	<b>54</b>
2.7.1. Cas de l'algorithme génétique .....	54
2.7.2. Approche distribuée dans les RCD.....	56
<b>2.8. CONCLUSION .....</b>	<b>57</b>
<b>PARTIE II: CONTRIBUTIONS.....</b>	<b>59</b>
<b>3. PROPOSITION D'APPROCHES CENTRALISEES POUR LES HYBRIDATIONS: STRATEGIES ET RESULTATS .....</b>	<b>61</b>
<b>3.1. INTRODUCTION.....</b>	<b>61</b>
<b>3.2. CONTRIBUTION 1: PREMIERE HYBRIDATION DVAG .....</b>	<b>61</b>
3.2.1. Algorithme proposé .....	61
3.2.2. Simulation .....	69
3.2.3. Résultats et discussions.....	70
<b>3.3. CONTRIBUTION 2 : SECONDE HYBRIDATION TDABC .....</b>	<b>75</b>
3.3.1. Algorithme proposé .....	75
3.3.2. Simulation .....	87
3.3.3. Résultats et discussions.....	88
<b>3.4. C-DVAG vs C-TDABC .....</b>	<b>91</b>
3.4.1. Taux de couverture .....	91
3.4.2. Coût de temps de calcul .....	92
3.4.3. Calcul des valeurs RSSI, durées de vie et nombres de voisins .....	93
<b>3.5. CONCLUSION .....</b>	<b>93</b>
<b>4. APPROCHE DISTRIBUEE : PROTOCOLE DE COMMUNICATION, SIMULATIONS, EXPERIMENTATIONS ET RESULTATS .....</b>	<b>95</b>
<b>4.1. INTRODUCTION.....</b>	<b>95</b>
<b>4.2. APPROCHE DISTRIBUEE POUR DVAG .....</b>	<b>96</b>
4.2.1. Stratégie de la distribution .....	96
4.2.2. Protocole de communication.....	96
4.2.3. Conception de l'algorithme de positionnement distribué.....	99
4.2.4. Résultats et discussions.....	101
<b>4.3. APPROCHE DISTRIBUEE DE D-TDABC.....</b>	<b>111</b>
4.3.1. Stratégie de distribution .....	111
4.3.2. Protocole de communication.....	112
4.3.3. Conception de l'algorithme de positionnement distribué.....	113
4.3.4. Résultats et discussions.....	115
<b>4.4. D-DVAG vs D-TDABC .....</b>	<b>125</b>
4.4.1. Taux de couverture .....	125



4.4.2. Coût de temps de calcul.....	125
4.4.3. RSS, durées de vie et nombres des voisins.....	126
<b>4.5. APPROCHE CENTRALISEE VS L'APPROCHE DISTRIBUEE.....</b>	<b>127</b>
4.5.1. Taux de couverture.....	127
4.5.2. Coût du temps de calcul .....	128
4.5.3. Comparaison en termes de consommation d'énergie .....	129
<b>4.6. CONCLUSION.....</b>	<b>130</b>
<b>CONCLUSION GENERALE ET PERSPECTIVES .....</b>	<b>131</b>
<b>PUBLICATIONS (DANS CETTE THESE).....</b>	<b>135</b>
<b>BIBLIOGRAPHIE .....</b>	<b>137</b>
<b>ANNEXES .....</b>	<b>153</b>

## LISTE DES FIGURES

Figure 0. 1 Les défis rencontrés dans un réseau de collecte des données .....	2
Figure 1. 1 Architecture général d'un réseau sans fil de collecte des données .....	10
Figure 1. 2 Les différents types de capteurs .....	11
Figure 1. 3 Portée de communication et portée de détection d'un nœud de capteur.....	14
Figure 1. 4 Architecture général d'un nœud capteur .....	15
Figure 1. 5 Classifications de RCD.....	15
Figure 1. 6 Architecture LoRaWAN .....	20
Figure 1. 7 NB-IoT Architecture .....	21
Figure 1. 8 Topologies IEEE 802.15.4 .....	22
Figure 1. 9 Un résumé des technologies de commutation sans fil .....	27
Figure 2. 1 Modèle de détection binaire.....	31
Figure 2. 2 Probabilité de couverture du point p ayant des différentes valeurs de $\omega$ .....	32
Figure 2. 3 Problème de couverture statique .....	33
Figure 2. 4 Classification des différentes méthodes de couverture .....	36
Figure 2. 5 Grille hexagonale .....	39
Figure 2. 6 Diagramme de Voronoi .....	41
Figure 2. 7 Triangulation et Triangulation de Delaunay d'un graphe de réseau non-planaire. ....	42
Figure 2. 8 Catégories d'algorithmes d'optimisation.....	44
Figure 2. 9 Organigramme de l'algorithme Génétique.....	47
Figure 2. 10 Organigramme de l'algorithme ABC.....	49
Figure 2. 11 Types d'Algorithme Génétiques Parallèle.....	56
Figure 3. 1 Diagramme de Voronoi.....	62
Figure 3. 2 Exemple d'une solution d'un déploiement initial des objets par le DV .....	63
Figure 3. 3 Suppression des solutions de Voronoi qui ont plus de deux nœuds en chevauchement .....	63
Figure 3. 4 Représentation d'un chromosome dans un champ 200*200 .....	64
Figure 3. 5 Représentation de la population initiale .....	65
Figure 3. 6 Chevauchement des nœuds.....	66
Figure 3. 7 Calcul d'aire des nœuds qui se chevauchent .....	67
Figure 3. 8 Croisement par un point .....	68
Figure 3. 9 Opérateur de mutation en un point.....	69
Figure 3. 10 Opérateur de mutation en deux points.....	69
Figure 3. 11 Déploiement aléatoire initial en utilisant DV .....	71
Figure 3. 12 Déploiement des nœuds après l'exécution de DVAG .....	71

Figure 3. 13 Comparaison du degré de couverture en fonction du nombre d'itérations pour DV, AG et DVAG .....	72
Figure 3. 14 Comparaison entre l'opérateur de mutation à un point et à deux points .....	73
Figure 3. 15 Générer N points aléatoire et définir les sommets du super triangle .....	77
Figure 3. 16 Processus de calcul de Triangulation de Delaunay.....	78
Figure 3. 17 Exemple de triangulation dans un champ 80*80.....	79
Figure 3. 18 Exemple de Triangulation de Delaunay dans un champ 80*80 .....	79
Figure 3. 19 Triangulation de Delaunay avec la portée de détection de chaque nœud .....	80
Figure 3. 20 Dépassement de nombre des nœuds (2 solutions TD).....	80
Figure 3. 21 Déploiement initiale de 20 nœuds par la TD (2 solutions) .....	81
Figure 3. 22 Les phases de l'algorithme ABC.....	82
Figure 3. 23 Une solution de déploiement de quatre nœuds dans un espace 2D (20*20).....	83
Figure 3. 24 Une solution de déploiement de quatre nœuds dans un espace 3D (20*20*4).....	83
Figure 3. 25 Produire une nouvelle solution.....	84
Figure 3. 26 Déploiement aléatoire initial en utilisant TD .....	88
Figure 3. 27 Déploiement des nœuds après l'exécution de 650 itérations de TDABC .....	89
Figure 3. 28 Comparaison du degré de couverture en fonction du nombre d'itérations pour TD, ABC et TDABC .....	89
Figure 3. 29 Comparaison du degré de couverture entre C-DVAG et C-TDABC .....	92
Figure 3. 30 Comparaison du coût de temps entre DVAG et TDABC .....	92
Figure 4. 1 Architecture distribuée proposée pour DVAG .....	96
Figure 4. 2 Architecture de réseau proposée.....	98
Figure 4. 3 Diagramme de séquence de notre protocole de communication entre trois nœuds .....	99
Figure 4. 4 Fonctionnement de l'approche proposée .....	100
Figure 4. 5 Déploiement de 20 nœuds avec DV seulement (exemple de 3 solutions) .....	103
Figure 4. 6 Déploiement de 20 nœuds après l'exécution de D-DVAG (exemple de 3 solutions).....	103
Figure 4. 7 Déploiement de 20 nœuds dans un champ 80*80 par DVAG [198].....	104
Figure 4. 8 Fonctionnement d'AG dans chaque Nœud.G.....	105
Figure 4. 9 Degré de couverture dans l'approche distribuée ayant six nœuds .....	106
Figure 4. 10 Les nœuds M5StickC ESP32 utilisés [200] .....	107
Figure 4. 11 Environnement d'expérimentation de DVAG .....	109
Figure 4. 12 Architecture distribuée proposée pour TDABC .....	111
Figure 4. 13 Communication entre un Nœud.TD et un Nœud.ABC.....	112
Figure 4. 14 La forme générale d'une sous-population de solutions TD.....	113
Figure 4. 15 Fonctionnement d'approche proposée.....	114
Figure 4. 16 Déploiement de 20 nœuds avec TD seulement (exemple de 3 solutions) .....	117
Figure 4. 17 Déploiement de 20 nœuds après l'exécution de TDABC (exemple de 3 solutions).....	118
Figure 4. 18 Déploiement de 20 nœuds dans un champ 80*80 par TDABC .....	118

Figure 4. 19 Degré de couverture dans l'approche distribuée pour six nœuds .....	119
Figure 4. 20 Environnement n°1 d'expérimentation .....	121
Figure 4. 21 Environnement n°2 d'expérimentation de TDABC .....	123
Figure 4. 22 Comparaison du degré de couverture entre DVAG et TDABC dans une approche distribuée.....	125
Figure 4. 23 Comparaison du coût de temps de calcul entre DVAG et TDABC dans une approche distribuée.....	126
Figure 4. 24 Comparaison du degré de couverture entre l'approche distribuée et l'approche centralisée .....	128
Figure 4. 25 Comparaison du coût de calcul de temps entre l'approche distribuée et l'approche centralisée .....	129

## LISTE DES TABLEAUX

Tableau 1. 1 Tableau comparatif entre les technologies sans fil utilisées pour RCD.....	26
Tableau 2. 1 Comparaison entre les différentes techniques basées sur la grille et sur la force virtuelle 40	
Tableau 2. 2 Hybridation de Voronoi avec des algorithmes d'optimisations ACO, PSO et GA .....	53
Tableau 3. 1 Paramètres de simulation de VDAG .....	70
Tableau 3. 2 Meilleure, moyenne et pire valeurs d'Hyper-volume de l'algorithme génétique .....	73
Tableau 3. 3 Comparaison des valeurs moyennes du RSSI, de la durée de vie et du nombre de voisins .....	74
Tableau 3. 4 correspondance entre les paramètres de l'algorithme ABC, et du déploiement des nœuds capteurs.....	83
Tableau 3. 5 Exemple sur le processus de sélection gloutonne .....	85
Tableau 3. 6 Exemple de calcul de probabilité d'une population de six solutions.....	85
Tableau 3. 7 Exemple de calcul de nombre d'essai (trial [i]) d'une population de six solutions .....	87
Tableau 3. 8 Paramètres de simulation de TDABC .....	88
Tableau 3. 9 Meilleure, moyenne et pire valeurs d'Hyper-volume de l'algorithme ABC .....	90
Tableau 3. 10 Comparaison des valeurs moyennes du RSSI, de la durée de vie et du nombre de voisins pour TD, ABC et TDABC.....	90
Tableau 3. 11 Comparaison des valeurs moyennes des métriques réseau pour les différents algorithmes .....	93
Tableau 4. 1 Paramètres de simulation de D-DVAG pour le Nœud.V .....	102
Tableau 4. 2 Paramètres de simulation de D-DVAG pour chaque Nœud.G .....	102
Tableau 4. 3 Les paramètres d'expérimentation de DVAG.....	107
Tableau 4. 4 Valeurs moyennes du nombre de voisins pour 6 nœuds .....	110
Tableau 4. 5 Valeurs moyennes de durée de vie (en secondes) pour 6 nœuds .....	110
Tableau 4. 6 Valeurs moyennes du nombre de voisins pour 6 nœuds .....	110
Tableau 4. 7 Paramètres de simulation de TDABC pour Nœud.TD.....	116
Tableau 4. 8 Paramètres de simulation de TDABC pour lesNœud.ABC.....	117
Tableau 4. 9 Les paramètres de premier cadre d'expérimentation .....	119
Tableau 4. 10 Emplacements des nœuds dans le premier cadre d'expérimentation.....	120
Tableau 4. 11 Les paramètres du deuxième cadre d'expérimentation .....	122
Tableau 4. 12 Emplacements des nœuds dans le deuxième lien d'expérimentation .....	123
Tableau 4. 13 Valeurs moyennes du RSSI pour 6 nœuds .....	124
Tableau 4. 14 Valeurs moyennes de durée de vie (en secondes) pour 6 nœuds .....	124
Tableau 4. 15 Valeurs moyennes du nombre de voisins pour 6 nœuds .....	124

Tableau 4. 16 Valeurs moyennes du RSSI pour 6 nœuds .....	126
Tableau 4. 17 Valeurs moyennes de durée de vie (en secondes) pour 6 nœuds.....	126
Tableau 4. 18 Valeurs moyennes du nombre de voisins pour 6 nœuds .....	127
Tableau 4. 19 Niveau de charge de batterie .....	129

## LISTE DES ALGORITHMES

Algorithme 3- 1 Pseudo-code de DVAG centralisée .....	62
Algorithme 3- 2 Pseudo code de TDABC centralisée.....	76
Algorithme 3- 3 Pseudo code de la phase des abeilles éclaireuses .....	86
Algorithme 4- 1 Pseudo-code de DVAG distribuée.....	101
Algorithme 4- 2 Pseudo-code de TDABC distribuée.....	115

## LISTE DES ACRONYMES

<b>2D</b>	Two-Dimensional, <i>Bidimensionnel</i>
<b>3D</b>	Three-Dimensional, <i>Tridimensionnel</i>
<b>3GPP</b>	3rd Generation Partnership Project, <i>Projet de partenariat de 3e génération</i>
<b>ABC</b>	Artificial Bee Colony, <i>Colonie d'abeilles artificielles</i>
<b>ACO</b>	Ant Colony Optimization, <i>Optimisation des colonies de fourmis</i>
<b>ADR</b>	Adaptive Data Rate, <i>Débit de données adaptatif</i>
<b>AG</b>	Algorithme génétique
<b>AHVGB</b>	Approximate to Horizontal and Vertical Grid Barrier, <i>Approximative à la barrière de grille horizontale et verticale</i>
<b>AoA</b>	Angle of Arrival, <i>Angle d'arrivée</i>
<b>AoD</b>	Angle of Departure, <i>Angle de départ</i>
<b>ASK</b>	Amplitude-Shift Keying, <i>Incrustation par décalage d'amplitude</i>
<b>BFO</b>	Bacterial Food Search Optimization, <i>Optimisation de la recherche de nourriture bactérienne</i>
<b>BLE</b>	Bluetooth Low-Energy, <i>Bluetooth à basse consommation</i>
<b>BPSK</b>	Binary phase-shift keying, <i>Modulation par déplacement de phase binaire</i>
<b>C-DVAG</b>	Centralisée- Diagramme de Voronoi Algorithme Génétique
<b>CDS</b>	Connected Dominating Set, <i>Ensemble dominant connecté</i>
<b>C-TDABC</b>	Centralisée- Triangulation Delaunay Artificial Bee Colony
<b>CSMA/CA</b>	Carrier Sense Multiple Access with Collision Avoidance, <i>Accès multiple avec écoute de la porteuse</i>
<b>CSS</b>	Chirp Spread Spectrum, <i>Spectre d'étalement</i>
<b>CSS</b>	Charged Search System, <i>Système chargé de recherche</i>
<b>DQPSK</b>	Differential Quadrature Phase Shift Keying, <i>Modulation par déplacement de phase en quadrature différentielle</i>
<b>DSSS</b>	Direct-sequence spread spectrum, <i>Etalement de spectre à séquence directe</i>
<b>DV</b>	Diagramme de Voronoi
<b>DVAG</b>	Diagramme de Voronoi Algorithme Génétique
<b>DVFA</b>	Distributed Virtual Forces- Algorithm, <i>Algorithme de force virtuelle distribuée</i>
<b>D-DVAG</b>	Distribuée- Diagramme de Voronoi Algorithme Génétique
<b>D-TDABC</b>	Distribuée - Triangulation Delaunay Artificial Bee Colony
<b>ED</b>	End-Devices, <i>Terminaux</i>
<b>EPC</b>	Electronic Product Code, <i>Code produit électronique</i>



<b>FCC</b>	Federal Communications Commission, <i>Commission fédérale des communications</i>
<b>FER</b>	Frame Error Rate, <i>Taux d'erreur de trame</i>
<b>FFD</b>	Full Function Device, <i>Appareil à fonctions complètes</i>
<b>FOA</b>	Fruit Fly Optimization, <i>Optimisation des mouches des fruits</i>
<b>FoI</b>	Field of Interest, <i>Champ d'intérêt</i>
<b>FSK</b>	Frequency Shift Keying, <i>modulation par déplacement de fréquence</i>
<b>GA</b>	Genetic Algorithm, <i>Algorithme génétique</i>
<b>GFSK</b>	Gaussian Frequency Shift Keying, <i>Modulation par déplacement de fréquence gaussienne</i>
<b>GPS</b>	Global Positioning System, <i>Système de positionnement global</i>
<b>GSO</b>	Group Search Optimization, <i>Optimiseur de recherche de groupe</i>
<b>GTS</b>	General Trading & Services, <i>Commerce général et services</i>
<b>HSA</b>	Harmony Search Algorithm, <i>Algorithme de recherche d'harmonie</i>
<b>IoT</b>	Internet of Things, <i>Internet des Objets</i>
<b>IR</b>	Infra-Rouge
<b>ISM</b>	Industrial, Scientific and Medical, <i>Industriel, scientifique et médical</i>
<b>KH</b>	Krill Herd, <i>Algorithme du troupeau de krill</i>
<b>LE</b>	Low Energy, <i>Basse énergie</i>
<b>LoRaWAN</b>	LoRa Wide Area Network, <i>LoRa réseau étendu</i>
<b>LPWAN</b>	Low Power Wide Area Network, <i>Réseau étendu à faible consommation</i>
<b>LR-WPAN</b>	Low Rate WPAN, <i>WPAN à bas débit</i>
<b>NB-IoT</b>	Narrow Band Internet of Things, <i>Internet des objets à bande étroite</i>
<b>OFDMA</b>	Orthogonal Frequency-Division Multiple Access, <i>Accès multiple par répartition orthogonale de la fréquence</i>
<b>OFDM</b>	Orthogonal Frequency Division Multiplex, <i>Multiplexage par répartition orthogonale de la fréquence</i>
<b>OQPSK</b>	Offset Quadrature Phase Shift Keying, <i>Modulation en quadrature de phase à décalage</i>
<b>PAN</b>	Personal Area Network, <i>Réseau personnel</i>
<b>PCA</b>	Probabilistic Coverage Algorithm, <i>Algorithme de couverture probabiliste</i>
<b>PBA</b>	Population Based Algorithm, <i>Algorithme basé sur la population</i>
<b>PGA</b>	Parallel Genetic Algorithm, <i>Algorithme génétique parallèle</i>
<b>PSK</b>	phase-shift keying, <i>modulation par changement de phase</i>
<b>PSO</b>	Particle Swarms Optimization, <i>Optimisation des essaims de particules</i>

<b>QPSK</b>	Quadrature Phase Shift Keying, <i>Manipulation de la phase de quadrature</i>
<b>RCD</b>	<i>Réseaux de Collecte des Données</i>
<b>RFA</b>	Rivers Formation Dynamics <i>Dynamique de formation des rivières</i>
<b>RFD</b>	Reduced Functional Device, <i>Les périphériques à fonction réduite</i>
<b>RFID</b>	Radio frequency identification, <i>Systèmes d'Identification par Radiofréquence</i>
<b>RoI</b>	Region of Interest, <i>Région d'intérêt</i>
<b>RSSI</b>	Received Signal Strength Indication, <i>indicateur de puissance du signal reçu</i>
<b>SDR</b>	Software Defined Radio, <i>Une radio logicielle</i>
<b>SF</b>	Spreading Factor, <i>Facteur d'étalement</i>
<b>SKH</b>	Stud Krill Herd, <i>Algorithme du troupeau de krill</i>
<b>SSA</b>	Salp–Swarm Optimizer, <i>Algorithmes de planification du sommeil</i>
<b>SSC</b>	Stud Selection and Crossover, <i>opérateur de sélection et de croisement des étalons</i>
<b>TD</b>	Triangulation de Delaunay
<b>TDABC</b>	Triangulation Delaunay Artificial Bee Colony
<b>ToF</b>	Time of Flight, <i>Temps de Vol</i>
<b>ULB</b>	Ultra Narrow Band, <i>Bande ultra étroite</i>
<b>UWB</b>	Ultra WideBand, <i>Bande ultra-large</i>
<b>VEC</b>	VECTor-based Algorithm, <i>Algorithme à base vectorielle</i>
<b>VFA</b>	Virtual Force Algorithm, <i>Algorithme de force virtuelle</i>
<b>VOR</b>	VORonoi-based Algorithm, <i>Algorithme à base de Voronoi</i>
<b>WLAN</b>	Wireless Local Area Network, <i>Réseau local sans fil</i>
<b>WPAN</b>	Wireless Personal Area Network, <i>Réseau personnel sans fil</i>
<b>WSP</b>	Wireless Short-Packet, <i>Paquet court sans fil</i>
<b>WSN</b>	Wireless Sensor Network, <i>Réseau de capteurs sans fil</i>



## Introduction générale

### Contexte et problématique

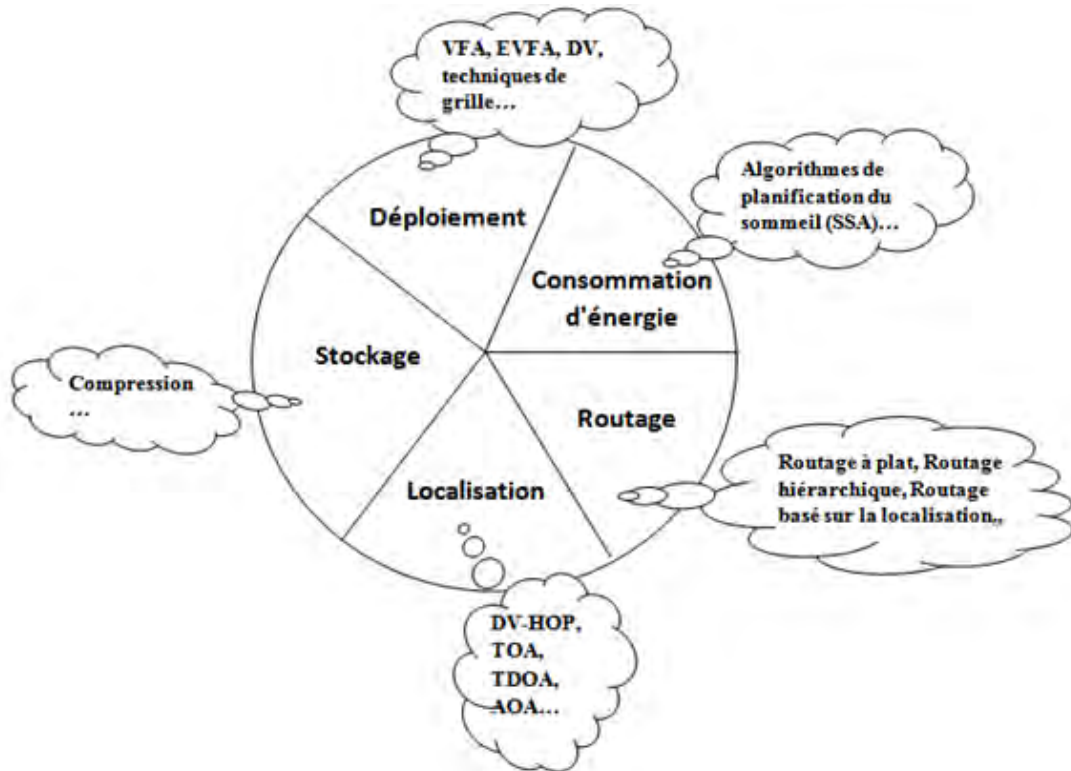
Le progrès de la technologie sans fil a eu de vastes répercussions au cours des dernières décennies, avec en particulier, l'amélioration de la portée de communication et l'évolutivité des réseaux sans fil. Parmi les différentes branches et versions des réseaux sans fil, les réseaux de collecte des données (**RCD**) ont reçu une attention particulière de la part de la communauté des chercheurs et du monde industriel par la suite. La raison en est leur large éventail d'applications en lien avec la récente révolution de l'Internet des objets (**IoT**). Par conséquent, la communication dans les RCD a connu des améliorations significatives ces dernières années, permettant leur déploiement dans des régions au terrain complexe ou dans des zones difficiles d'accès. Les RCD se composent généralement d'un ensemble de nœuds capteurs qui captent et suivent les changements de facteurs environnementaux tels que la température, la salinité de l'eau et l'humidité. Chaque capteur a une plage de détection connue, dans laquelle les données environnementales sont collectées en continu, à des intervalles de temps variables suivant la dynamique des phénomènes à capter.

Les réseaux sans fil de collecte de données sont largement utilisés pour collecter les informations nécessaires à la création des environnements intelligents, de bâtiments et de villes intelligents, de services publics, d'industries, de maisons, de navires, de systèmes de transport ou autres. Les **RCD** peuvent être utilisés aussi pour la sécurité des infrastructures dans les bâtiments et dans les installations critiques, telles que les centrales électriques et les centres de communication. Les industries commerciales s'intéressent depuis longtemps à la surveillance comme moyen efficace pour réduire les coûts et d'améliorer les performances et la maintenabilité des machines. La surveillance des machines et de leur "état santé" est rendue possible par la détermination des niveaux de vibration, ou d'usure ou de lubrification, et le rajout de capteurs dans des régions inaccessibles aux humains [1].

Une classification élargie des applications de réseau sans fils de collecte de données peut inclure la surveillance continue de phénomènes, par exemple pour évaluer la santé et les conditions de croissance des plantes, ou pour observer et mesurer des processus géophysiques, la détection d'événements en temps réel tel que l'inondation et les volcans.

Dans un réseau sans fil de collecte des données, tous les nœuds sont déployés dans des environnements divers. Ces nœuds capteurs qui utilisent les données localement ou à travers la communication avec d'autres réseaux, fonctionnent en coopération pour acquérir et rassembler des informations de la zone surveillée via des liaisons sans fil. L'utilisation de RCD dans des divers domaines est un succès technologique et les chercheurs y voient un progrès prometteur pour l'avenir [2].

Malgré le développement continu des RCD, il existe encore plusieurs défis de communication entre les nœuds, comme le schématise la figure 1. Il est nécessaire de réduire le coût matériel des nœuds capteurs et augmenter de la durée de vie du réseau. Par conséquent, un routage sécurisé [3, 4], une efficacité énergétique [5, 6], des techniques de traitement et de compression de données pour le stockage [7, 8], et des choix de la bonne stratégie de déploiement [9, 10] sont des défis importants dans les RCD.



**Figure 0. 1 Les défis rencontrés dans un réseau de collecte des données**

L'un de ces défis est la détermination efficace des positions des capteurs dans l'environnement, également connu sous le nom de problème de déploiement ou de problème de placement des nœuds capteurs.

Dans cette thèse de recherche nous abordons le problème de placement de nœud capteurs dans un champ d'intérêt bien déterminé.

Trouver les meilleurs emplacements de capteurs pour obtenir la couverture souhaitée d'une région d'intérêt, et assurer en même temps une bonne qualité des communications sans fil, est un problème important dans le déploiement d'un réseau de capteurs. Dans plusieurs articles issus de la littérature, la couverture est définie en fonction de la visibilité entre l'observateur et les points cibles [11]. Plus précisément, dans les réseaux sans fil de collecte des données, la couverture d'un point signifie que le point est situé dans le champ de détection d'un nœud capteur. Le non-respect de cette condition pour certains points de la région d'intérêt entraînera des trous de

couverture. Sur la base de cette définition mentionnée de la couverture dans les réseaux de capteurs, le problème de couverture réside dans le placement d'un minimum de nœuds dans un environnement, de sorte que chaque point d'intérêt dans le champ de détection soit bien couvert [12,13].

### **Objectifs de recherche et principales contributions**

Les réseaux sans fil de collecte de données ont été intensément étudiés par les chercheurs dans les dernières années. De nombreux travaux de recherche ont été réalisés et de nombreux articles ont été publiés, mais de nombreux aspects méconnus restent encore à étudier [14, 15, 16, 17]. L'objectif général de notre thèse est donc d'améliorer la couverture spatiale dans le déploiement **3D**, principalement à l'intérieur, d'un réseau sans fil de collecte de données, et donc par conséquent, il est nécessaire de trouver le meilleur emplacement d'un nœud capteur pour atteindre une couverture souhaitée, dans une région d'intérêt. Ainsi, la définition d'un cadre conceptuel pour le déploiement d'un réseau sans fil de collecte de données est un enjeu important de l'objectif général de cette thèse.

Différentes approches heuristiques peuvent être envisagées pour optimiser les RCD. La nature distribuée et dynamique du problème de déploiement nécessite l'utilisation des méthodologies d'optimisation avancées et de modélisation : les méta-heuristiques hybrides. Ces méthodologies d'optimisation permettent d'exploiter la complémentarité de ces méthodes entre elles et de profiter des avantages d'autres approches conventionnelles hybridées avec elles. Cette nouvelle classe de méta-heuristiques hybrides a montré sa performance dans la résolution de problèmes d'optimisation difficiles notamment face à des objectifs qui concernent principalement l'assurance d'un système évolutif, flexible et robuste face aux perturbations du réseau.[18].

Les travaux présentés dans cette thèse de recherche se proposent d'apporter six contributions principales :

1. Une première hybridation DVAG entre une méthode géométrique de déploiement présentée par le Diagramme de Voronoi avec un algorithme d'optimisation présenté par l'algorithme génétique, est introduite dans une approche centralisée où l'exécution se fait dans un seul nœud. Le but est d'améliorer le taux de couverture dans les réseaux de collecte de données avec le minimum de nœuds capteurs.
2. Une deuxième hybridation TDABC entre une méthode géométrique de déploiement présentée par le diagramme de Triangulation de Delaunay avec un algorithme d'optimisation présenté par l'algorithme de colonie des abeilles artificielles ABC est introduite dans une approche centralisée où l'exécution se fait dans un seul nœud. L'objectif est d'améliorer le taux de couverture

dans les réseaux de collecte de données avec le minimum des nœuds capteurs.

3. Le traitement de certaines parties algorithmiques de la charge de travail dans des différentes ressources processeurs disponibles sur l'ensemble des nœuds du réseau, pourrait résoudre le problème de charge de traitement. Nous introduisons une approche distribuée pour les hybridations DVAG et TDABC. Notre but principal est de partager les tâches aux différents nœuds utilisés dans un réseau afin de gagner du temps et limiter la consommation de chaque nœud. L'exécution de ces tâches se fait en parallèle sur les différents nœuds utilisés pour le déploiement dans un réseau sans fil de collecte de données.
4. Un protocole de communication avec une stratégie de distribution a été utilisé pour implémenter l'approche distribuée dans les deux hybridations. Ce protocole de communication a pour but de gérer la communication entre les différents nœuds utilisés pour le déploiement.
5. Les approches DVAG et TDABC distribuées, ont été évaluée dans un environnement expérimental par prototypage sur des bancs d'essai réels à l'aide de nœuds M5StickC. A notre connaissance, ce type de nœud est utilisé pour la première fois pour résoudre les problèmes de déploiement dans un réseau sans fil de collecte de données.
6. Le dernier travail consiste alors à comparer et mesurer les résultats obtenus avec l'approche centralisée et l'approche distribuée appliquées aux deux hybridations DVAG et TDABC. Chaque comparaison se fait en termes de taux de couverture, coût de temps de calcul, consommation énergétique et avec des métriques telles que le RSSI , la durée de vie et le nombre des voisins de chaque nœud.

### **Organisation de la thèse**

Ce manuscrit s'articule autour de quatre chapitres répartis en deux parties. Une première partie concerne l'état de l'art comportant les deux premiers chapitres et une deuxième partie qui concerne les contributions proposées comporte les chapitres 3 et 4.

Nous commençons dans le premier chapitre par exposer en premier lieu, les principales terminologies utilisées dans les réseaux sans fil de collecte des données telles que nœud capteur, nœud mobile, nœud redondant, liens de communication et les type des RCD existants. En deuxième lieu nous présentons l'architecture d'un nœud capteur générique et les unités qui le composent. En troisième lieu, nous proposons un classement des réseaux sans fil de collecte des données tel que les RCD terrestres et les RCD mobiles. Nous aboutissons à la fin de ce chapitre à la présentation des technologies émergentes utilisées récemment pour les RCD et leurs

aspects techniques respectifs y compris la technologie radio protocolaire et architecturale utilisée pour implémenter nos approches de déploiement.

Dans le deuxième chapitre, nous présentons tout d'abord les notions de couverture et de connectivité dans un réseau sans fil de collecte des données. Ensuite, nous nous intéresserons à présenter le problème de déploiement, pour lequel les nœuds capteurs doivent être positionnés avec précision dans des emplacements bien déterminés pour réaliser un ou plusieurs objectifs. Nous citons un grand nombre d'études qui ont proposé des algorithmes pour résoudre le problème de déploiement. Puis nous nous intéressons par la suite à présenter une étude comparative entre les approches d'optimisation récentes qui sont utilisées pour résoudre le problème de déploiement dans les RCD. Dans cette même partie nous étudions le Diagramme de Voronoi **DV** et la Triangulation de Delaunay **TD** comme méthodes géométriques de déploiement et les algorithmes d'optimisations que nous utiliserons dans nos approches de déploiement tel que l'algorithme génétique **AG** et l'algorithme de colonie des abeilles artificielles **ABC**. Nous finissons par une présentation des études récentes procédant par des hybridations entre les méthodes géométriques et les algorithmes d'optimisation pour résoudre le problème de déploiement.

Dans le troisième chapitre, nous présentons notre approche centralisée maximisant la couverture d'une région d'intérêt par l'application des deux hybridations. La première hybridation **DVAG** combine une méthode géométrique de déploiement présentée par le Diagramme de Voronoi **DV** avec un algorithme d'optimisation présenté par l'algorithme génétique **AG**. La deuxième hybridation **TDABC** combine une méthode géométrique de déploiement présentée par le diagramme de Triangulation de Delaunay **TD** avec un algorithme d'optimisation présenté par l'algorithme de colonie des abeilles artificielles **ABC**. Nous présentons par la suite les résultats obtenus pour chaque hybridation. Nous finissons ce chapitre par une comparaison entre **DVAG** dans une approche centralisée **C-DVAG** et la deuxième hybridation centralisée **C-TDABC**. Cette comparaison est faite en analysant le taux de couverture obtenu, le coût de temps de calcul et les métrique **RSSI** et durée de vie.

Dans le quatrième chapitre, nous présentons notre approche distribuée pour **DVAG** et **TDAB** où le traitement se fait dans plusieurs nœuds en parallèle. L'objectif est d'atteindre la solution souhaitée dans les plus brefs délais avec une meilleure performance par rapport à une approche centralisée. Ce chapitre commence tout d'abord par l'approche distribuée pour notre première hybridation **D-DVAG** où nous présentons la stratégie de distribution, le protocole de communication entre les nœuds ainsi que les résultats obtenus. Ensuite, nous traitons de l'approche distribuée pour la deuxième hybridation **D-TDABC** ainsi que les résultats obtenus. L'approche proposée est évaluée pour les deux hybridations sur un environnement expérimental par prototypage sur un banc d'essai réel à l'aide de nœuds **M5StickC** équipés d'un processeur **ESP32**. Puis nous comparons les deux hybridations **DVAG** et **TDABC** dans l'approche distribuée. Et nous nous attachons à la fin de ce chapitre à comparer



les performances atteintes entre les résultats obtenus par l'approche centralisée (chapitre 3) avec les résultats obtenus par l'approche distribuée dans ce chapitre. Enfin, nous présentons nos conclusions avec un rappel portant sur les différentes contributions réalisées tout au long de ce travail de recherche, et mettons en relief les perspectives de recherche futures.

# **Partie 1 : État de l'art**

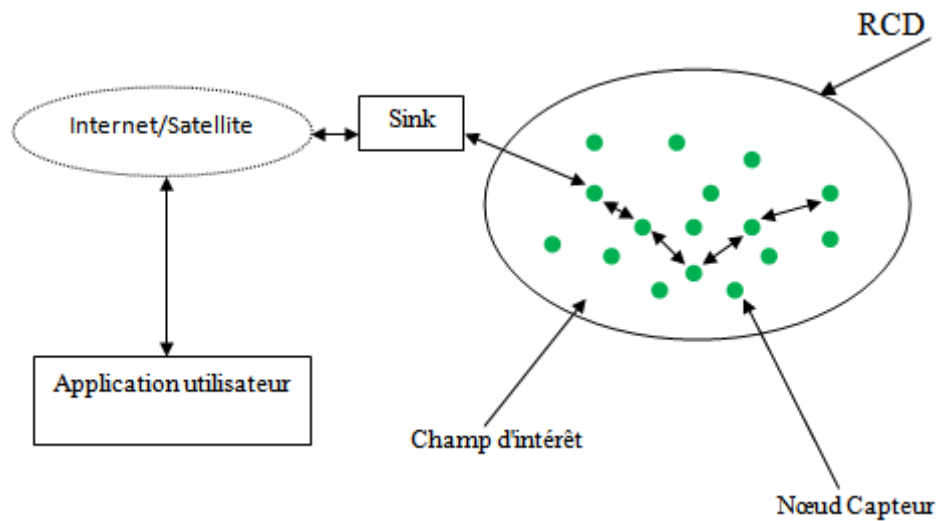
# **1. Les réseaux sans fil de collecte de données de l'IoT: problématique et état de l'art**

## 1.1. Introduction

Selon [19], d'ici 2025, jusqu'à 75 milliards d'appareils seront connectés à l'Internet des objets (IoT). L'Internet des objets joue un rôle crucial dans la réalisation d'un suivi, d'un positionnement, d'une identification, d'une surveillance et d'une gestion de l'environnement intelligent. Grâce à des capteurs, à des dispositifs de lecture de codes bidimensionnels, des dispositifs d'identification par radiofréquence (RFID), à des systèmes de positionnement global (GPS), des scanners laser, des stockages en Cloud et d'autres équipements réseaux, l'IoT relie les choses du monde physique et du cybermonde et rend la vie humaine plus pratique et confortable.

Un réseau sans fil de collecte de données (RCD) est un élément indispensable de l'IoT, qui est utilisé par de nombreuses applications [20]. Jour après jour, de nombreuses nouvelles applications sont également inventées utilisant de ce système. L'expression « réseau sans fil de collecte des données » suggère que l'ensemble des nœuds présents dans le réseau sont des nœuds capteurs qui sont utilisés pour détecter et contrôler l'environnement qui les entoure [21]. Les nœuds du réseau ont la capacité de communiquer entre eux. Nous savons que les capteurs sont utilisés pour détecter tout changement d'environnement. Par exemple, un capteur peut être capable de détecter la pression atmosphérique, la température, des sons, tout mouvement, etc. Les réseaux de capteurs sans fil peuvent être définis comme un réseau auto-créé avec des nœuds capteurs eux-mêmes. De nos jours, la popularité des RCD augmente en raison de leurs applications quasi illimitées. Dans tous les secteurs, nous pouvons trouver l'utilisation de RCD comme dans la gestion et le contrôle de l'environnement [22], dans le secteur de la santé et du médical, la localisation, le suivi, la surveillance, dans le domaine militaire, pour une maison intelligente ou une ville intelligente [23,24], etc.

Les nœuds capteurs d'un RCD communiquent avec d'autres nœuds connectés et agissent sur les informations qu'ils obtiennent les uns des autres. La connectivité, la mise en réseau, et les protocoles de communication utilisés avec ces nœuds dépendent fortement des applications RCD spécifiques mises en œuvre. Bien que les personnes puissent interagir avec ces nœuds, comme la configuration, le paramétrage ou l'accès aux données, l'appareil effectue la majeure partie du travail sans intervention humaine [25].



**Figure 1. 1 Architecture général d'un réseau sans fil de collecte des données**

Dans ce chapitre, nous exposons tout d'abord dans la section 2 les principales terminologies utilisées dans les réseaux sans fil de collecte des données tel que capteur, lien de communication... Dans la troisième section, nous présentons l'architecture d'un nœud capteur générique et les unités qui le composent. Nous finirons ce chapitre en citant les technologies émergentes utilisées récemment pour les RCD et leurs aspects techniques respectifs.

## **1.2. Notions préliminaires sur le réseau de collecte de données**

Cette section présente les principales définitions, les objectifs, les stratégies et les limitations majeures pour le réseau de collecte de données. Pour comprendre les stratégies de déploiement en détail, certaines terminologies doivent être clarifiées.

### **1.2.1. Capteurs**

Tous les réseaux sans fil de collecte des données doivent avoir au moins un capteur pour recueillir des informations de l'environnement [26]. Les capteurs sont des éléments de base des objets connectés ou « smart ». La plupart de capteurs sont de petite taille, nécessitent une capacité de calcul minimal et consomment peu d'énergie. Différents types de capteurs utilisés dans les applications IoT sont présentés dans la Figure 1.2



Figure 1. 2 Les différents types de capteurs [27]

De nombreux capteurs peuvent être utilisés pour détecter des données dans un RCD tel que le capteur infrarouge (capteur **IR**), capteur température, capteur à ultrasons qui serve à détecter et mesurer des distances entre divers types d'objets, quelle que soit leur forme (liquide, solide, granuleux, etc.), capteur de lumière, couleur etc.

### 1.2.2. Capteurs mobiles

Examinons le Smartphone, qui est de plus en plus universel et intègre de nombreux types de capteurs. Avec l'importance croissante des téléphones portables chez les individus, les scientifiques manifestent leur intérêt pour la construction d'architectures IoT intelligentes utilisant des téléphones portables en raison des capteurs installés [28]. Certains capteurs supplémentaires peuvent également être utilisés en fonction du contexte. Les applications peuvent être basées sur le téléphone qui utilise les informations des capteurs pour fournir des résultats significatifs [29].

### 1.2.3. Lien de communication

Généralement après avoir terminé la configuration du système matériel et du logiciel d'un appareil et l'avoir préparé, vient l'étape de la communication, qui est chargée de fournir le lien de transmission des appareils. Le système de communication est basé sur des systèmes protocolaires logiciels et matériels, son importance ne doit pas être négligée, car elle a une forte importance dans les réseaux sans fil [30]. Tout d'abord, nous devons décider de la technologie de transmission des données, nous avons plusieurs choix disponibles sur le marché. Ces choix peuvent inclure WiFi,

Bluetooth, ZigBee, LoRa, etc...Chacun d'eux est adapté à un type transmission en fonction des matériels et des logiciels respectifs installés. Chaque transceiver radio ou puce de communication a sa configuration, et non seulement elle décidera du mode d'envoi, mais également de la manière avec laquelle les appareils sont gérés et communiquent. Nous verrons à la dernière section de ce chapitre les technologies émergentes utilisées récemment pour les RCD pour la communication.

#### **1.2.4. RCD hétérogène**

Les nœuds utilisés dans ce type de réseau sont capables de réaliser des traitements multiples; par exemple, limiter la consommation d'énergie, la distribution des données, etc. Ces capteurs sont beaucoup plus complexes que les autres types pour gérer la consommation d'énergie et la durée de vie [31].

#### **1.2.5. RCD homogène**

Les nœuds utilisés dans ce type de réseau ont tous les mêmes capacités de stockage, de détection, de traitement, de puissance et de capacité de communication. Un RCD homogène équipés de nœuds identiques augmente le risque que les nœuds qui gèrent de nœuds (cluster) soient surchargés par de nombreuses transmissions. Par conséquent, cela nécessite une coordination des protocoles et une agrégation des données. Cependant, selon [32] ils sont assez faciles à manipuler.

#### **1.2.6. Nœuds relais**

Les nœuds relais sont différents des autres nœuds parce qu'ils nécessitent un espace de stockage plus grand que les autres types des nœuds pour accueillir plus de données. Ils sont également responsables de l'accumulation des données détectées et retransmises vers le puits via d'autres nœuds relais ou têtes de cluster [33].

#### **1.2.7. Nœuds redondants**

Ce sont les nœuds similaires qui ne sont pas utilisés dans le réseau pour la détection de données captées sur l'environnement. Cela signifie que s'ils sont supprimés, ceci n'affectera pas le réseau. Ils sont utiles au cas où certains nœuds ont des pannes, ils prennent alors la place des nœuds défaillants.

#### **1.2.8. Nœuds mobiles**

Ils sont similaires aux autres nœuds mais ont des capacités de mobilité, ils peuvent donc changer de place selon les besoins. Ils jouent un rôle important en cas d'amélioration de la connectivité, de la couverture, ainsi qu'en cas de défaillance de certains nœuds. Une tâche de récupération ou cicatrisation de la topologie du réseau peut également être réalisée lorsque le nœud mobile fonctionne comme un routeur dans une zone de couverture faible ou inexistante.

#### **1.2.9. Puits mobiles**

Ceux-ci sont similaires aux nœuds mobiles mais contrairement à la détection des données, ils recueillent uniquement les informations détectées par les autres nœuds

capteurs. Ces données sont ensuite utilisées par l'utilisateur. Ils sont bénéfiques car ils prolongent la durée de vie du réseau et aident à l'équilibrage de charge. La consommation d'énergie est également réduite significativement [34].

#### **1.2.10. Notion de trou d'énergie**

Cette situation se produit en raison de la consommation irrégulière de la charge de données sur les nœuds. Étant donné que les nœuds les plus proches du puits font à face à un trafic de données plus conséquent, ces derniers sont donc plus susceptibles de tomber en panne énergétique. Cela peut être corrigé en déployant davantage de nœuds redondants au niveau du puits [35].

### **1.3. Architecture d'un nœud capteur**

Les entités de base d'un réseau de collecte de données sont des nœuds capteurs. Un nœud capteur comprend une unité de détection, de traitement, de stockage et de communication [36]. Le fonctionnement d'un réseau se fait en trois phases: détection, traitement et transmission au récepteur en passant par des nœuds intermédiaires [37] plus au moins nombreux. L'unité de détection d'un nœud capteur s'interface avec son environnement physique pour acquérir les informations requises. Les transmissions des trames ou paquets de données et de contrôle à travers le réseau sont prises en charge par l'unité de communication d'un nœud [38]. Les nœuds capteurs sont également dotés d'une unité de traitement qui peut réaliser des opérations algorithmiques sur ces données. Cette unité est généralement constituée d'un microprocesseur ou microcontrôleur. Le nœud est également associé à une unité de mémoire qui est utilisée pour stocker un ensemble d'instructions et des données détectées [39].

#### **1.3.1. Unité de détection**

Cette entité est chargée de capter les informations de son environnement. Elle répond au stimulus physique souhaité, tels que pression, humidité, température, etc. Selon le principe de fonctionnement, les nœuds capteurs sont généralement divisés en deux types : les nœuds de capteurs actifs et les nœuds capteurs passifs. Les nœuds de capteurs actifs sont utilisés pour mesurer des données physiques environnementales trop faibles ou qui nécessitent un stimulus [40]. Ce type de nœud capteur a besoin d'une énergie supplémentaire pour générer et obtenir le signal. Les nœuds de capteurs passifs surveillent la force du signal directement émis par l'environnement physique. Ici les capteurs passifs ne nécessitent pas d'alimentation supplémentaire [41]. Il convient de noter que les nœuds de capteurs passifs consomment moins d'énergie que les nœuds de capteurs actifs.

Nous pouvons définir une portée de détection comme étant la distance maximale entre un nœud et un emplacement présent dans un champ d'intérêt (**RoI**) donné à l'intérieur de laquelle un nœud capteur peut détecter un événement qui se produit. Le nœud capteur doit couvrir la plage de détection du nœud. Le réseau **2D** facilite la

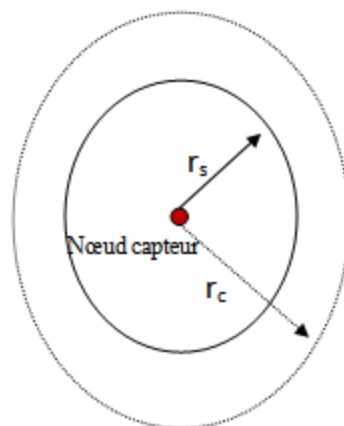


modélisation des contours en forme de disque de la zone d'inspection, alors que l'on modélise souvent par la zone de détection pour un réseau de collecte des données sans fil 3D . Cette plage peut donc être exprimée en termes de rayon d'un disque ou d'une sphère [42].

### 1.3.2. Unité de communication

Cette unité est responsable de la communication entre les nœuds. Grâce à cette unité, les paquets de contrôle et de données sont transférés entre les nœuds des réseaux. La communication entre les nœuds capteurs a été classifiée en communication bidirectionnelle ou unidirectionnelle.

Nous pouvons définir ici une portée de communication par la séparation euclidienne maximale entre un nœud et un autre nœud qui permet aux nœuds de communiquer. Inversement, deux nœuds capteurs ne communiqueront entre eux que lorsque la séparation euclidienne entre ces deux nœuds est inférieure à la portée de communication de ces nœuds.



**Figure 1. 3** Portée de communication ( $r_c$ ) et portée de détection( $r_s$ ) d'un nœud de capteur

### 1.3.3. Unité de traitement

Cette unité exécute certaines tâches, traite des données et contrôle l'état de fonctionnement des autres modules d'un nœud. Selon la tâche, différents types de processeurs peuvent être utilisés, tels que petits microcontrôleurs assez complets, des microprocesseurs plus performants, des processeurs de communication et d'autres circuits intégrés spécifiques à une application conçus pour des tâches spécifiques [43].

### 1.3.4. Unité de stockage

Il s'agit d'une partie importante d'un nœud capteur, qui peut être utilisée pour stocker un ensemble d'instructions et des informations sensorielles. Généralement un nœud capteur est muni d'une batterie limitée qui alimente tous ses composants

électroniques comme le montre la figure 1.4. L'énergie est la ressource la plus précieuse car elle affecte directement la durée de vie des nœuds capteurs.

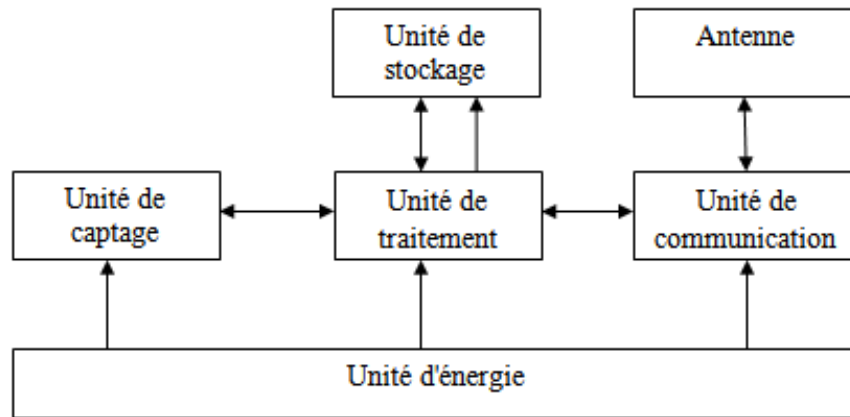


Figure 1. 4 Architecture général d'un nœud capteur

#### 1.4. Classement de réseau sans fil de collecte des données

Le réseau sans fil de collecte des données dispose de nombreux types de capteurs en fonction de la situation tels que les RCD souterrains, les RCD sous-marins, les RCD terrestres, les RCD multimédias et les RCD mobiles, comme le schématise la figure 1.5.

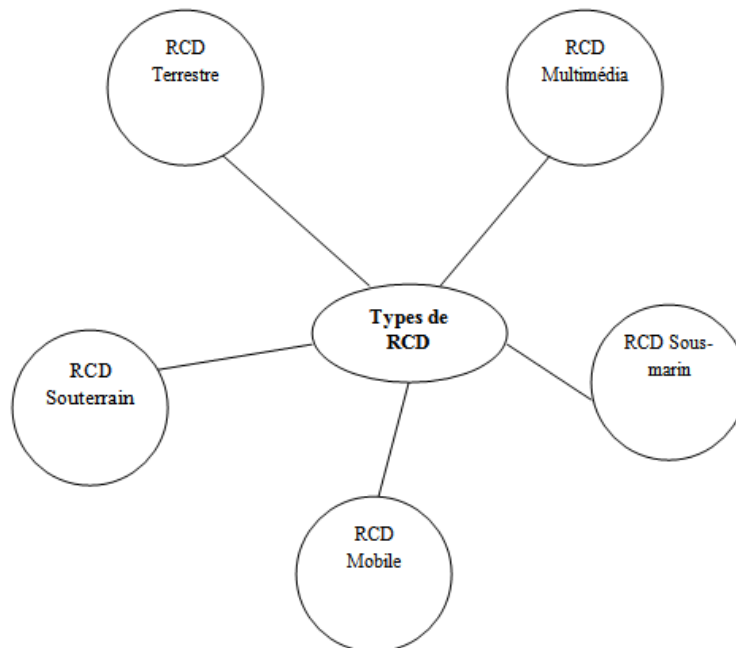


Figure 1. 5 Classifications de RCD

#### **1.4.1. Les RCD Terrestres**

Les RCD terrestres sont généralement constitués de nombreux nœuds de capteurs qui sont déployés de manière aléatoire (à partir d'un avion par exemple) dans une zone spécifique à partir de laquelle un réseau ad hoc est utilisé pour la communication entre les nœuds. Ces nœuds peuvent être organisés par placement optimal, placement de grille ou placement **2D** et **3D** modèles [44]. Les auteurs dans [45] ont étudié les applications du capteur terrestre sans fil dans les systèmes radiofréquence/optique dans les espaces libres. Les lacunes des applications terrestres sont l'effet des conditions météorologiques telles que la neige et la pluie sur une liaison de communication sans fil optique. Les auteurs ont proposé une méthode pour augmenter les performances de durée de vie du réseau.

#### **1.4.2. Les RCD Souterrains**

Dans un RCD souterrain, les nœuds capteurs sont placés sous le sol dans des caves, des conduits, des égouts ou sous la terre pour collecter des informations sur les conditions souterraines. L'inconvénient des RCD est lié à la capacité des batteries utilisées dans les nœuds qui sont limitées et pour lesquelles il est très difficile de les recharger ou les changer. Le principal inconvénient de la communication sans fil est l'affaiblissement du signal radio lors de la traversée des obstacles. Dans le but d'améliorer la fiabilité du RCD, des algorithmes de routage efficaces ont été mis en œuvre pour résoudre cette limitation pendant le processus de routage [46]. Les auteurs [47] ont proposé un cadre mathématique afin d'analyser la connectivité dynamique dans le réseau de capteurs souterrains sans fil. L'objectif était de surveiller les changements dans les paramètres environnementaux tels que l'humidité aléatoire du sol et la composition du sol. Généralement ces types de réseau sont exploités dans l'agriculture, le suivi des réseaux d'eau potable les mines souterraines, la sismologie.

#### **1.4.3. Les RCD sous-marins**

Un RCD sous-marin comprend des nœuds capteurs assez coûteux, qui sont placés sous l'eau (comme les mers et océans) pour collecter des données et des informations sur l'ambiance sous-marine. Certaines limitations dans la communication sans fil souterraine incluent le grand délai de propagation sous l'eau, la bande passante très réduite, et des forts affaiblissements du signal. Ce problème a été reconnu par les chercheurs qui, et afin d'éradiquer le problème, ont proposé une solution via un algorithme de routage adapté [48]. Les auteurs [49] ont proposé un réseau de capteurs sous-marins sans fil basé sur la communication optique entre nœuds. Les fréquences radio et les ondes acoustiques sont profondément atténuées dans l'eau. Pour surmonter ce problème, la communication optique peut être envisagée. Généralement, les réseaux sous-marins sont exploités dans l'exploration des fonds marins, la robotique sous-marine, la détection de la pollution, la prévention des maladies et des agents chimiques...

#### **1.4.4. Les RCD mobiles**

Un nœud d'un RCD mobile a l'avantage de pouvoir changer de position et de se connecter automatiquement à l'environnement. Des capteurs mobiles sont utilisés pour collecter des données à partir d'une zone étendue ou des informations provenant d'autres nœuds ou capteurs. Les principales problématiques des RCD mobiles sont de bien appréhender l'état courant des nœuds, d'assurer le repositionnement des nœuds, la maintenance des capteurs, d'optimiser la zone de couverture et d'assurer une navigation optimales des mobiles. Ces problèmes ont été en partie résolus via un algorithme de routage dynamique [50].

#### **1.4.5. Les RCD multimédias**

Un RCD multimédia se compose de plusieurs nœuds capteurs qui peuvent détecter, calculer, actionner et communiquer des données binaires, audio, vidéos... Il existe une grande variété d'applications des RCD qui comprend la surveillance des habitations, la surveillance écologique et la gestion du trafic des systèmes. Le RCD multimédia est un réseau de nœuds interconnectés sans fil appareils capables de récupérer de manière omniprésente des flux de transmission vidéo et audio, les images fixes et de simples données des capteurs de l'environnement [51]. Les auteurs dans [52] ont analysé les développements récents sur les RCD multimédias et décrit les principaux défis de la recherche. Des algorithmes, protocoles et matériel pour l'amélioration des capteurs multimédias ont été étudiés et des analyses ouvertes ont été menées pour les problèmes évoqués. Les auteurs ont tendance à classer le matériel prêt à l'emploi comme un matériel accessible pour réaliser un prototype de recherche pour les réseaux multimédia.

### **1.5. Exigences d'un réseau de collecte des données sans fils**

Nous présentons ici les principales caractéristiques et exigences d'un système comprenant des nœuds de capteurs sans fil. Le système doit disposer des caractéristiques suivantes

#### **1.5.1. Tolérance aux pannes**

Le système doit être robuste contre les pannes de nœud tel que le manque d'énergie etc... Des alertes sonores devraient être prévues par exemple pour indiquer qu'un nœud ne fonctionne pas correctement.

#### **1.5.2. Évolutif**

Le système doit prendre en charge un grand nombre de nœuds de capteurs pour répondre à différentes applications. Le système doit s'adapter à un ajout de nouveaux capteurs, à la complexité de la topologie du réseau.

#### **1.5.3. Longue durée de vie**

La durée de vie du nœud induit fortement la durée de vie du réseau qui doit être suffisamment élevée. Le nœud capteur doit être économe en énergie par rapport aux

ressources énergétiques limitées dont il dispose car il est difficile pour remplacer ou recharger des centaines de nœuds. Les calculs, les traitements, les communications, les détections et toutes les autres opérations doivent également être économes en énergie.

#### **1.5.4. Programmable**

La reprogrammation du capteur nœuds sur le terrain pourrait être nécessaire pour améliorer la flexibilité, et ainsi pouvoir faire des mises à jour à travers le réseau lui-même (Over the Air).

#### **1.5.5. Sécurité**

Le nœud doit prendre en charge de nombreux éléments de sécurité :

- Contrôle d'accès : pour empêcher les tentatives d'accès non autorisées au nœud,
- Intégrité du message : pour détecter et empêcher les modifications non autorisées du message,
- Confidentialité : pour garantir que le nœud capteur doit pouvoir chiffrer les messages afin de garantir que seuls les nœuds autorisés puissent écouter et comprendre les messages en les décryptant par des clés secrètes.
- Protection : pour garantir que le nœud capteur puisse fournir une protection contre la réutilisation par l'adversaire d'un paquet authentique pour gagner confiance/réseau l'accès, éviter l'attaque bien connue de « l'homme du milieu » qui peut être évitée par des paquets de données horodatées. Ceci implique de pouvoir disposer d'une horloge unique sur tous les nœuds du réseau.

#### **1.5.6. Abordable**

Le système doit utiliser un système à faible coût pour les appareils car le réseau peut comprendre des centaines de nœuds capteurs. La mise en place et la maintenance des éléments du système doit également être significativement faible pour rendre son déploiement réaliste.

### **1.6. Les technologies utilisées dans les réseaux de collecte des données sans fil**

Dans cette section, nous allons étudier les technologies émergentes utilisés récemment dans les réseaux sans fil de collecte des données et leurs aspects techniques respectifs, cette étude fait l'objet de nos publications [53, 54].

#### **1.6.1. LoRa**

LoRa a été développé pour la première fois par la start-up Cycleo en 2009 (à Grenoble, en France) et a été acheté trois ans plus tard par Semtech (États-Unis). En 2015, LoRa a été normalisé par la LoRa-Alliance qui l'a déployé dans 42 pays. Ce déploiement est toujours en cours dans plusieurs pays du fait des investissements de

différents opérateurs mobiles (Bouygues et Orange en France, KPN aux Pays-Bas et Fastnet en Afrique du Sud, par exemple) [55].

LoRa est une technologie de couche physique qui module les signaux dans la bande ISM sub-GHZ en utilisant une technique propriétaire à spectre étalé [56]. LoRa utilise des bandes ISM sans licence, soit 915 MHz en Amérique du Nord, 868 MHz en Europe et 433 MHz principalement en Asie. La communication bidirectionnelle est fournie par la modulation CSS qui traite un signal à bande étroite sur une largeur de bande de canal très grande. Le signal résultant a un niveau de bruit faible, permettant une résilience élevée aux interférences, il est difficile à détecter ou à bloquer [57].

LoRa utilise six facteurs d'étalement (**SF**) pour adapter par compromis le débit de données et d'intervalle spatial. Un facteur d'étalement plus élevé permet une portée plus longue au détriment d'un débit de données plus faible, et inversement. Le débit de données de la couche physique LoRa est compris entre 300 bps et 50 kbps en fonction du facteur d'étalement et de la largeur de bande du canal. De plus, les messages transmis en utilisant différents facteurs de propagation peuvent être reçus simultanément par les stations de base LoRa [58]. La taille maximale de la charge utile pour chaque message est de 243 octets. Le protocole de communication réseau basé sur la couche radio LoRa appelé **LoRaWAN** a été normalisé par la LoRa-Alliance dans sa première version en 2015. Avec LoRaWAN, chaque message transmis par un terminal est potentiellement reçu par plusieurs stations de base à portée. En exploitant cette réception redondante, LoRaWAN améliore la qualité de réception de messages. Cependant, cette fonctionnalité nécessite plusieurs stations de base dans le voisinage, ce qui peut augmenter les coûts de déploiement du réseau.

Un réseau LoRaWAN utilise une topologie de réseau en étoile, dans laquelle une passerelle relaie de manière transparente les messages entre un serveur de réseau (**NS**) et les appareils finaux **ED** (End-Devices), comme le montre la figure 1.6 Les ED utilisent la couche radio LoRa pour communiquer avec les passerelles (**GW**). Les GW utilisent un réseau IP (Ethernet, 3G, 4G, WiFi, etc.) pour communiquer avec le serveur de réseau.

La communication entre les appareils et les passerelles est répartie sur différents canaux de fréquences et les débits de données sont déterminés en fonction de la portée de communication et de la durée du message. Cette sélection peut être gérée par une infrastructure de réseau LoRaWAN, qui sélectionne le débit de données et le canal pour chaque périphérique à l'aide d'un schéma **ADR** (Adaptive Data Rate) [59].

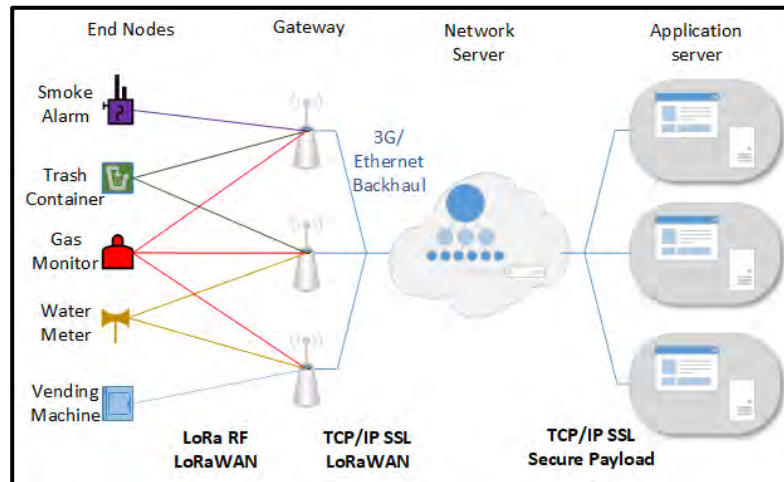


Figure 1. 6 Architecture LoRaWAN [60]

### 1.6.2. SigFox

Sigfox propose une solution de connectivité IoT de bout en bout basée sur ses technologies brevetées. Sigfox déploie ses stations de base propriétaires équipées de radios cognitives définies par logiciel (SDR) et les connecte aux serveurs principaux à l'aide d'un réseau IP (3G, 4G, Internet, WiFi...). Les équipements terminaux connectés à ces stations de base utilisent une modulation **BPSK** (binary phase-shift keying) avec une porteuse dans une sous-bande ISM ultra-étroite (100 Hz). Sigfox utilise des bandes ISM sans licence, pour exemple, 868 MHz en Europe, 915 MHz en Amérique du Nord et 433 MHz en Asie. En utilisant la bande **ULB** (Ultra Narrow Band), Sigfox exploite efficacement la bande passante et présente des niveaux de bruit très faibles, ce qui entraîne une consommation d'énergie très faible, une sensibilité élevée du récepteur et la conception d'antenne à faible coût, au détriment d'un débit maximal de 100 bits/s seulement. Initialement, Sigfox ne prenait en charge que la communication montante, mais a ensuite évolué vers la technologie bidirectionnelle avec un lien asymétrique offrant quelques bits dans la voie descendante vers les objets connectés. La communication de la liaison descendante, c'est-à-dire des données des stations de base aux équipements terminaux ne peut se produire qu'après une communication en liaison montante (comme **LoRaWAN**). Le nombre de messages sur la liaison montante est limité comme LoRaWAN par un rapport cyclique très réduit, typiquement à 140 messages SigFox par journée avec une longueur maximale de la charge utile pour chaque message de 12 octets. De plus, le nombre de messages sur la liaison descendante est limité à 4 messages par jour, ce qui signifie par la suite que l'accusé de réception de chaque liaison montante le message n'est pas supporté. La longueur maximale de la charge utile pour chaque message de la liaison descendante est de huit octets. La fiabilité de la communication sur la liaison montante est assurée à l'aide de la diversité de temps et de fréquences ainsi que de la duplication de la transmission. [61,62]

### 1.6.3. NB -IoT

Le réseau Narrow Band Internet of Things (NB-IoT) est une nouvelle technologie LPWAN basée sur la technologie radio à bande étroite et normalisée par le projet de partenariat de 3ème génération (3GPP). L'architecture générale de la technologie NB-IoT est illustrée à la figure.1.7 [63, 64]. Les différentes entités de l'architecture sont décrites comme suit :

- 1) UE NB-IoT : cette entité établit la connexion et communique en sans fil avec un eNodeB.
- 2) eNodeB : il a pour tâche d'effectuer le traitement des accès de l'interface radio et de la cellule la gestion. Il communique via l'interface S1-lite avec l'IoT Evolved Packet Core (EPC).
- 3) EPC : ce sont des interfaces EPC IoT avec le NAS de l'UE qui acheminent des données IoT vers la plate-forme IoT pour traitement ultérieur.
- 4) Plate-forme IoT : elle accumule les données IoT de divers réseaux d'accès IoT et transmet différentes données à chaque serveur d'applications.
- 5) Serveur d'application : le serveur d'application est le point final d'agrégation des données. En outre, il est le destinataire final de l'IoT et traite les données conformément aux spécifications et aux exigences du client.

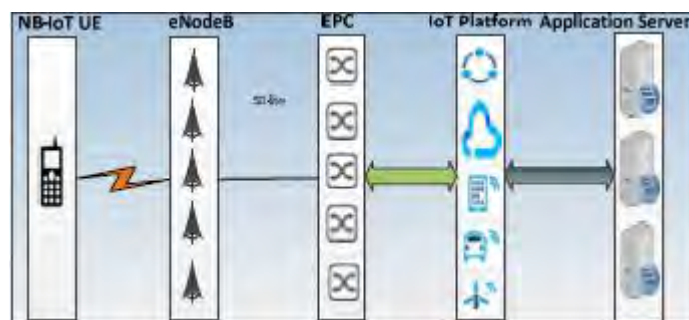


Figure 1. 7 NB-IoT Architecture [63]

### 1.6.4. IEEE 802.15.4

Après la présentation des 3 principales technologies LPWAN ci-dessus, nous allons nous focaliser maintenant sur les WLAN et WPAN. Le protocole IEEE 802.15.4 a été créé pour spécifier une sous-couche pour le contrôle d'accès moyen (MAC) et une couche physique (PHY) pour les réseaux privés sans fil à faible débit (LR-WPAN) [65].

Ce protocole de niveau 1 et 2 est la base du protocole ZigBee qui offrent tous les deux des services à faible débit de données sur des périphériques à faible consommation d'énergie en construisant une pile de protocoles réseau complète pour les RCD. Les réseaux IEEE 802.15.4 prennent en charge trois bandes de canaux de fréquence et utilisent une méthode d'étalement de spectre étalé à séquence directe (DSSS). Dans les versions initiales, En fonction des canaux de fréquence utilisés, la couche physique transmet et reçoit des données avec trois débits de données : 20 kbps à 868 MHz, 40 kbps à 915 MHz et 250 kbps à 2,4 GHz. Les fréquences plus



élevées et des bandes plus larges fournissent une faible latence et un débit élevé tandis que les basses fréquences offrent une meilleure sensibilité et couvrent de plus grandes distances. IEEE 802.15.4 utilise principalement le protocole CSMA/CA avec pour objectif de réduire substantiellement les collisions potentielles. Une autre MAC est prévue, basée sur une allocation temporelle de slots de temps (GTS) mais elle est rarement implémentée.

La norme IEEE 802.15.4 prend en charge deux types de nœuds de réseau : les périphériques à fonctions complètes et à fonctions réduites. Le FFD (Full Function Device) peut servir de coordinateur de réseau personnel (PAN) ou simplement de nœud normal. Un nœud FFD coordinateur est responsable de la création, du contrôle et de la maintenance du réseau. Les nœuds FFD peuvent stocker une table de routage dans leur mémoire et implémenter une MAC complète. Ils peuvent également communiquer avec tout autre périphérique utilisant n'importe quelle topologie disponible. Les périphériques à fonction réduite (RFD), sont des nœuds très simples avec des ressources limitées. Ils ne peuvent communiquer qu'avec un coordinateur et sont limités à une topologie en étoile. Les nœuds RFD sont souvent les capteurs du réseau.

Les topologies classiques permettant de former des réseaux IEEE 802.15.4 sont les suivantes : étoile(a), peer to peer (b) et cluster-tree (c) comme le montre la figure 1.8. La topologie en étoile (a) contient au moins un FFD et quelques RFD. Le FFD qui travaille en tant que coordinateur PAN doit être situé au centre de la topologie et vise à gérer et contrôler tous les autres nœuds du réseau. La topologie peer to peer (b) contient un coordinateur PAN et d'autres nœuds qui communiquent entre eux dans le même réseau ou via des nœuds intermédiaires vers d'autres réseaux.

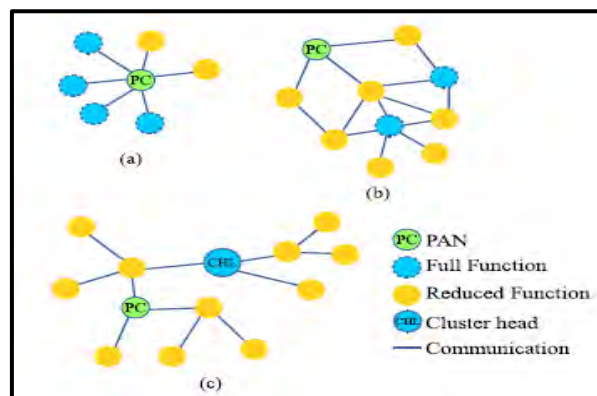


Figure 1. 8 Topologies IEEE 802.15.4 [65]

### 1.6.5. BLE

Bluetooth est un protocole de communication sans fil, a pour objectif de connecter des appareils mobiles entre eux.

Ici aussi, la consommation d'énergie est une exigence importante, la spécification LE (Low Energy) vise à fournir des communications sans fil à faible consommation et à faible coût. La spécification LE a été introduite pour la première fois dans la norme Bluetooth v4.0 [66] et amélioré ultérieurement dans la norme Bluetooth v4.2 [67]. La norme Bluetooth v5.0 [68] améliore encore la spécification LE du standard Bluetooth 4.2 (portée, vitesse et la capacité du message de radio-diffusion) [69]. Par conséquent, la spécification LE de la norme Bluetooth v5.0 est considéré comme une solution idéale de l'Internet des objets (IoT) et a été largement déployée dans des applications personnelles ou commerciales. Le récent standard Bluetooth v5.1 [70] conserve toujours la même conception Low Energy que la norme Bluetooth v5.0 avec une fonction supplémentaire appelée "Direction Finding" qui permet de déterminer la direction du signal Bluetooth. Cette version permet d'offrir des meilleures fonctionnalités de localisation, à travers l'angle d'arrivée AoA et l'angle de départ AoD en permettant de connaître la direction, ainsi que la force du signal pour obtenir une position plus précise.

Cette v5.1 intègre aussi un tracker d'activité associé aux beacons Bluetooth qui a pour objectif de réduire la consommation électrique de 5 à 10 fois par rapport au niveau standard, pour l'abaisser entre 10 et 100 microwatts, avec un débit de 1 Mb/s, en particulier lors des parcours personnalisés de visiteurs dans des musées.

Afin de maintenir cette consommation, une autre technologie permet de plonger l'appareil dans un profond état de veille, Un composant « écouteur » matériel très basse consommation se charge permettant d'intercepter les transmissions radio, sans émettre de paquets, afin de réveiller l'ensemble du nœud.

De plus, la v5.1 introduit le concept de "récolteur d'énergie", et permet d'utiliser l'énergie des ondes radio environnantes de 900 à 1 400 MHz afin de générer quelques microwatts et ainsi d'alimenter la puce elle-même plongée le reste du temps dans un état de veille. Ceci permet aux puces de fonctionner sans batterie ni pile.

Bluetooth Low-Energy ( BLE) ou Bluetooth Smart utilise une radio à courte portée avec une quantité d'énergie minimale pour fonctionner plus longtemps (même pendant des années) par rapport à ses versions précédentes. Sa portée d'environ 100 mètres est dix fois supérieure à celle du Bluetooth classique [66]. BLE peut fonctionner avec une puissance de transmission comprise entre 0,01 et 10 mW. Avec ces caractéristiques, BLE est un bon candidat pour les applications IoT [67].

La norme BLE a été développée rapidement par les fabricants de Smartphones et est maintenant disponible dans la plupart des modèles. La faisabilité de l'utilisation de cette norme a été démontrée dans des communications de véhicule à véhicule ainsi que de réseaux de capteurs sans fil.

#### **1.6.6. EnOcean**

EnOcean est une technologie de communication sans fil à ultra-basse consommation alimentée par la récupération d'énergie. La technologie EnOcean associe des convertisseurs micro-énergie à une électronique ultra-basse consommation, ce qui a

permis d'obtenir des communications sans fil fiables grâce à une pile de protocoles simple [71].

La spécification EnOcean standard comprend quatre couches : couche physique, couche liaison de données, couche réseau et couche application. La Norme internationale ISO / IEC 14543-3-10 spécifie les couches OSI de 1 à 3 du protocole **WSP** (Wireless Short-Packet) [72] et l'organisation ouverte et indépendante EnOcean Alliance normalise la couche d'application [73]. La couche physique régule le transfert des données sur la fréquence 315 MHz ou la bande de fréquence 868,3 MHz avec un débit binaire de 125 kbit/s en utilisant la modulation d'amplitude par décalage (**ASK**) [72]. La distance fonctionnelle du système est jusqu'à 300 m en visibilité directe. En couche de liaison de données, EnOcean utilise un schéma de contrôle d'accès direct (**MAC**). L'accès direct au support utilisé par EnOcean permet à un appareil de transmettre un message instantanément et de passer ensuite directement en mode veille [74]. Ce qui permet aux commutateurs EnOcean d'envoyer leurs messages avant que l'énergie récupérée en appuyant sur le commutateur ne soit épuisée. En utilisant des messages très courts, un mécanisme "Listen before Talk" et des répétitions de messages plusieurs fois, EnOcean réduit les risques de collision de messages et permet des économies d'énergie [75]. La couche réseau est responsable d'étendre le réseau EnOcean avec les répéteurs, gérer l'évitement des collisions, réaliser l'adressage des télégrammes EnOcean.

### **1.6.7. UWB**

La norme IEEE 802.15.4 intègre les spécifications d'une couche radio particulière utilisant une technique d'hyper étalement de spectre **UWB**. La technologie UWB offre une transmission à haute vitesse, une relative faible consommation énergétique, et une mise en œuvre simple. La bande passante élevée exploitable avec UWB permet le streaming vidéo en haute définition. UWB peut être utilisé pour un nombre élevé d'applications militaires, commerciales et grand public, y compris un certain nombre d'applications IoT. Parmi les applications traditionnelles de l'UWB, on peut citer l'imagerie médicale, la surveillance, les applications de radar de véhicule pour les systèmes de transport intelligents, les applications de localisation par mesure de temps To, les applications de mesure et les applications de communication.

Les capacités d'UWB permettent de fournir des liaisons de données relativement élevées avec une immunité contre le bruit et les interférences de trajets multiples, une faible consommation d'énergie, ainsi que des émetteurs-récepteurs simples et finalement peu coûteux, en font un candidat idéal pour les applications IoT à courte portée. La technologie peut être utilisée à l'intérieur des maisons, des bureaux, des véhicules de tous types et dans des établissements (hôpitaux, par exemple). Des exemples d'applications IoT basées sur UWB sont par exemple : les services d'annonce, les services d'assistance, le téléchargement de contenu (applications domestiques), l'assistance d'urgence, la recherche de victimes ou d'objets, la gestion

de flotte, l'instrumentation biomédicale des soins de santé connectivité domestique / Home cinéma, télémétrie hospitalière [76].

Les flux de transmission du système UWB sont extrêmement courts avec des impulsions de 10 à 10 000 ps réparties sur une partie du spectre RF à large bande. En raison des faibles niveaux d'émission autorisés par les divers organismes de réglementation, les systèmes UWB de communication sont généralement mieux adaptés aux applications intérieures à courte portée. Certaines de ces applications à l'intérieur comprennent la connectivité de périphérique (par exemple, les imprimantes sans fil, bien que celles-ci puissent également utiliser le WiFi), le transfert d'images à partir de caméscopes et de lecteurs multimédias [77].

Aux États-Unis, la Federal Communications Commission (FCC) a mis en place des bandes UWB sans licence offrant une bande passante de 7,5 GHz (de 3,1 GHz à 10,6 GHz) et des UWB à la fréquence d'onde millimétrique centrée à 60 GHz (de 57 GHz à 64 GHz). UWB évolue maintenant rapidement en tant que technologie de communication sans fil à large bande [76].

Finalement, le principal atout d'UWB face à ses concurrents est sa capacité à évaluer les distances entre les nœuds de façon très précises, ce qui permet la localisation des nœuds en intérieur et extérieur. Le mécanisme d'évaluation de la distance entre des nœuds (ranging) est essentiel pour identifier les emplacements de ces derniers de façon précise. Pour obtenir un ranging performant basé sur le temps de vol, **ToF** (Time of Flight), les auteurs [78] ont exploité les propriétés de la couche physique UWB. Ces propriétés fournissent des précisions de l'ordre de la dizaine de picoseconde sur les estampilles temporelles des émissions et réceptions de trames. Une précision de l'ordre de 10cm est donc atteinte. Les dernières versions de Smartphone (Apple, Samsung, Google...) sont maintenant équipés nativement de puces UWB, ce qui laisse envisager des applications de localisation pour par exemple lutter contre le piratage lors des paiements sans contacts, l'ouverture des portes par clés numériques...

### **1.6.8. WiFi**

Il s'agit d'une technologie sans fil populaire basée sur IEEE 802.11 et utilisée généralement dans les réseaux domestiques. La vitesse peut atteindre 7 Gbits/s avec la dernière version WiFi 6e. Son taux d'adoption est extrêmement élevé. Le Wifi peut être moins sécurisé que les connexions filaires [79]. Le WiFi a une consommation d'énergie élevée car les débits de données et la portée sont élevés. Un point d'accès sans fil utilisant 802.11g et 802.11n a une portée de 35 m à l'intérieur et de 100 m à l'extérieur. En comparaison avec le réseau de capteur sans fil populaire basé sur ZigBee, qui utilise IEEE802.15.4, le WiFi est plus mature et présente des avantages évidents dans l'interopérabilité des appareils. Le réseau WiFi repose sur la technologie de réseau IPv4 et IPv6, avec lesquelles le réseau WiFi distribué dans les entreprises et à la maison peut être facilement connecté, il a également une meilleure sécurité. Le réseau de capteurs sans fil basé sur le WiFi se compose de nœuds de

consommation d'énergie répartis dans la zone de détection, des nœuds puits et le réseau qui se construit par les méthodes d'auto-organisation. Les nœuds WiFi sont peu coûteux, miniatures et disposent de capacités de traitement de données et de communication auto-organisée. La consommation électrique des nœuds de capteurs WiFi est si faible qu'une pile AA suffit pour 5 à 10 ans [80], si bien sûr la quantité de données transmises est limitée. Les versions actuelles de WiFi utilisent les bandes 2.4Ghz et 5Ghz.

Comme nous le verrons dans les chapitres suivants, nous avons utilisé cette technologie pour la communication entre les différents nœuds de réseau sans fil de collecte de données pour résoudre le problème de déploiement à l'intérieur.

Le tableau suivant synthétise les principales caractéristiques des réseaux sans fils présentés ci-dessus.

**Tableau 1. 1Tableau comparatif entre les technologies sans fil utilisées pour les réseaux de collecte des données**

	LoRa	NB-IoT	SigFox	IEEE 802.15.4	BLE	EnOcean	UWB	WiFi
<b>Modulation</b>	CSS	QPSK	BPSK	OQPSK, DQPSK et BPSK	GFSK	ASK	BPSK QPSK	ASK; FSK; PSK , OFDM
<b>Fréquence</b>	868 MHz Europe 915 MHz USA 433 MHz Asie	4.33GHz, 915MHz, 2.4GHz to 5GHz	868 MHz Europe 915 MHz MHz USA 433 MHz Asie	2.4 GHz, 868 MHz, 915 MHz	2.4 GHz	868 MHz	3.1-10.6 GHz	2.4Ghz et 5Ghz
<b>Bande passante</b>	250KHz et 125 KHz	200 kHz	100 Hz	5 Mhz et 2 Mhz	2 Mhz	500 KHz	500 MHz	20 MHz, 40 MHz
<b>Débit des données (Maximum)</b>	50 kbps	200 kbps	100 bps	1 Mbs	500 kbps (coded) 2 Mbps (uncoded)	120kb/s	110 Mbits/s	7 Gbits/s
<b>Consommation d'énergie</b>	Faible	Faible	Faible	Faible	Très faible (5) Extrêmement Faible Autonomie, sans pile (5.1)	Extrêmement Faible Autonomie, sans pile	Haute	Haute

La figure 1.9 présente un résumé des technologies actuelles dans les réseaux sans fil de collecte de données. Les technologies de communication à faible consommation d'énergie pour la communication IoT sans fil peuvent grossièrement appartenir à

deux catégories: la première pour les réseaux personnels sans fil (WPANs) tel que BLE, ZigBee et WiFi, la deuxième catégorie pour les réseaux étendus à basse consommation (LPWAN) tel que LoRa et sigfox.

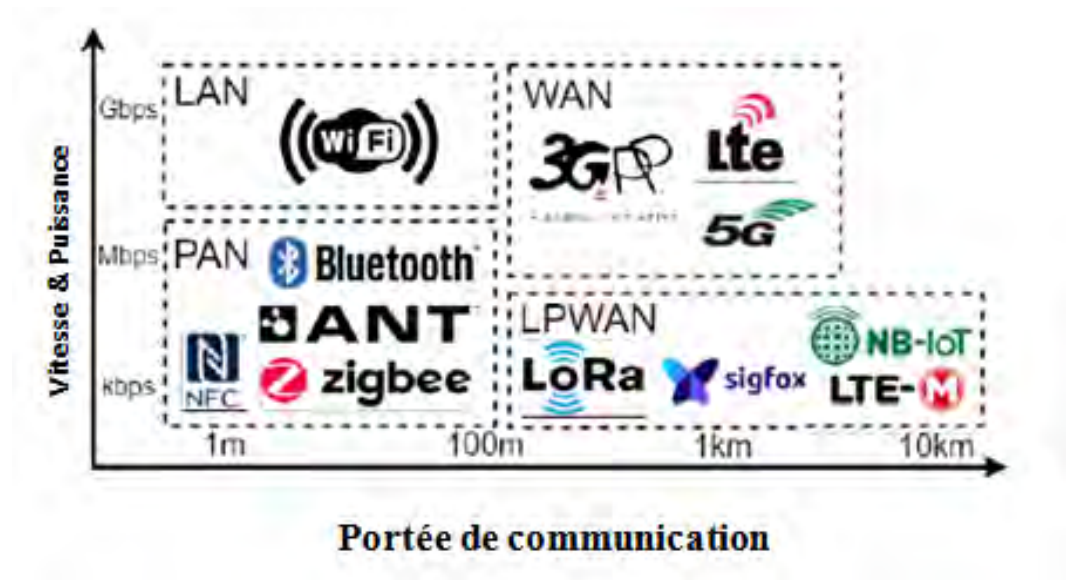


Figure 1. 9 Un résumé des technologies de commutation sans fil [81]

### 1.7. Conclusion

Dans ce chapitre, nous avons décrit ce qu'est un réseau sans fil de collecte de données et quelles sont ses principales caractéristiques. Nous avons présenté les bases et évoqué certains concepts nécessaires qui doivent être pris en compte dans la conception du réseau de capteurs. Nous nous sommes également concentré sur les technologies qui peuvent être utilisées dans un réseau sans fil de collecte des données de capteurs premier maillon de l'IoT.

## **2. Le problème de déploiement optimisé pour les réseaux de collecte de données**

## 2.1. Introduction

Le déploiement est un problème important marquant les **RCD** car il affecte directement les performances des réseaux. L'utilisation d'un grand nombre de nœuds de capteurs dans l'environnement imprévisible de déploiement améliore les problèmes d'évolutivité et de fiabilité [82]. Le déploiement de capteurs peut être non déterministe lorsqu'il s'agit d'une zone sinistrée, mais il est aussi question de déploiement déterministe où nous voulons placer des capteurs pour surveiller la température par exemple [83]. Comme évoqué au chapitre 1, alors que la stratégie de déploiement homogène utilise les mêmes nœuds, la stratégie de déploiement hétérogène utilise des nœuds ayant diverses fonctionnalités telles que la puissance de traitement et la plage de détection. Le déploiement dans un RCD affecte presque toutes les mesures de performance, telles que la connectivité entre les différents nœuds capteurs, la couverture et la durée de vie du réseau [84]. Le but de déploiement est de déterminer le type, les emplacements et le nombre des nœuds pour maximiser le taux de couverture, améliorer la connectivité et élever la durée de vie du réseau.

Lors de la conception des stratégies de déploiement, plusieurs facteurs doivent être pris en compte, tels que la zone de surveillance, la portée de détection et de transmission d'un nœud capteur ainsi que la zone de couverture globale du RCD et la durée de vie attendue. Dans ce deuxième chapitre, nous allons présenter tout d'abord la couverture et la connectivité dans un réseau sans fil de collecte des données. Ensuite nous nous intéresserons au problème de déploiement, dans lequel les nœuds capteurs doivent être positionnés avec précision dans des emplacements bien déterminés pour réaliser un ou plusieurs objectifs de conception dans un réseau sans fil de collecte des données. Il existe un grand nombre d'études qui ont proposé des algorithmes pour résoudre le problème de déploiement. Nous nous intéressons également dans ce chapitre à présenter une étude comparative entre les approches d'optimisation récentes utilisées pour résoudre le problème de déploiement dans les réseaux sans fil de collecte des données. Dans cette partie nous serons soucieux d'étudier le Diagramme de Voronoi **DV** et la Triangulation de Delaunay **TD** comme méthodes géométrique de déploiement et sur les algorithmes d'optimisations que nous utiliserons dans nos approches de déploiement tel que l'algorithme génétique **AG** et l'algorithme de colonie des abeilles artificielles **ABC**. Et nous nous intéressons à la fin de ce chapitre à exposer des études récentes procédant par hybridations et distributions des méthodes géométrique avec les algorithmes d'optimisation pour résoudre le problème de déploiement.

## 2.2. Modèles de couverture

Le modèle de couverture se concentre sur l'efficacité de la capacité à suivre par les événements à l'intérieur du champ d'intérêt (**RoI**). Ce modèle de couverture travaille



sur la liaison entre la séparation euclidienne entre la position d'un nœud capteur avec la cible et la sensibilité du nœud capteur [85].

Supposons qu'un point  $p$  est situé aux coordonnées  $(x,y)$  dans un champ d'intérêt et qu'un nœud  $n_i$  a les coordonnées ou l'emplacement  $(x_i,y_i)$ , la séparation euclidienne (ou distance) entre le capteur  $n_i$  et le point  $p$  est donnée dans l'équation. (1):

$$d(n_i, p) = \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (1)$$

Si le point  $p$  est situé aux coordonnées  $(x,y,z)$  dans un champ d'intérêt (3D) et qu'un nœud  $n_i$  a les coordonnées ou l'emplacement  $(x_i,y_i,z_i)$ , la séparation euclidienne entre le capteur  $n_i$  et le point  $p$  est donnée dans l'équation

$$d(n_i, p) = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2}$$

Selon la littérature, il y a d'autres paramètres tel que le classificateur K-plus proche voisin (KNN) [205] peut être utilisé, ce classificateur dépend principalement de la mesure de la distance ou de la similarité entre les exemples testés et les exemples d'apprentissage.

La sensibilité générale de  $n_i$  au point  $p$  est donnée dans l'équation. (2) :

$$(n_i, p) = \frac{\varphi}{d(n_i, p)^k} \quad (2)$$

$d(n_i, p)$  représente la séparation euclidienne entre le nœud  $n_i$  et le point  $p$ . De plus,  $k$  et  $\varphi$  sont des constantes non négatives dépendant du nœud capteur.

### 2.2.1. Modèle de couverture binaire

Le modèle de couverture binaire suppose que chaque capteur affiche une cellule ou disque avec un rayon égal à la distance de détection respective  $r_s$  comme déjà montré à la figure.1.3. Un nœud peut détecter uniquement les valeurs des points qui se trouvent à l'intérieur de son propre disque de détection respectif. Un modèle de couverture binaire peut être présenté mathématiquement dans l'équation. (3) :

$$C_{xy}(n_i) = \begin{cases} 1 & \text{sid}(n_i, p) < r_s \\ 0 & \text{autre} \end{cases} \quad (3)$$

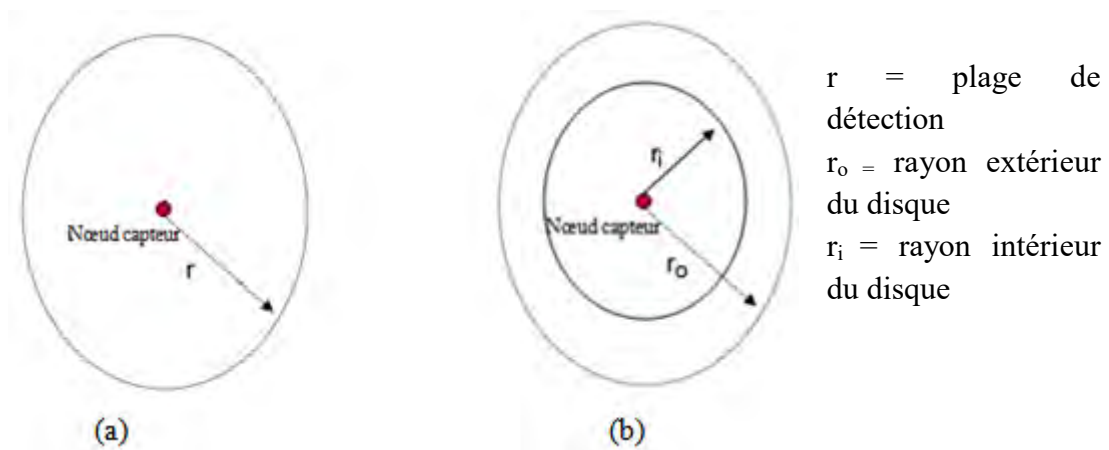
Avec  $C_{xy}(n_i)$  montre que  $n_i$  couvre le point  $p$ .

Selon Wang et al. [86], il existe deux types de modèles de détection binaires : les modèles de détection binaires parfaits et les modèles de détection imparfaits. Dans le

modèle de détection binaire parfait, chaque nœud est capable de reconnaître si la cible tombe dans sa plage de détection  $r$  comme illustré à la figure 2.1 (a)).

Dans le modèle de détection binaire imparfait ou fragile, la cible est toujours diagnostiquée dans le disque interne de rayon  $r_i$  comme elle est illustré à la figure 2.1 (b). Cependant ce modèle ne peut être découvert qu'avec une certaine probabilité non nulle dans un anneau entre ce disque intérieur et le disque extérieur de rayon  $r_o$ . Bien que le modèle de capteur binaire soit plus simple, le facteur d'incertitude de la mesure n'y est pas pris en compte. En réalité, les mesures par un nœud de capteur sont imprécises et il est probable que les signaux physiques passent soudainement de valeurs élevées de pleine intensité à zéro, comme le suppose le modèle binaire.

Cela signifie qu'il est possible de diagnostiquer des événements qui se produisent à des distances supérieures au rayon de détection.



(a) modèle de détection binaire parfait

(b) modèle de détection binaire imparfait

**Figure 2. 1 Modèle de détection binaire [86]**

### 2.2.2. Modèles de détection probabilistes

Un modèle de détection probabiliste est beaucoup plus pratique étant donné que l'occurrence détectée, la conception du capteur et les circonstances environnementales sont toutes de nature stochastique. Par exemple, le bruit et les interférences dans l'environnement naturel peuvent être modélisés par des processus stochastiques [87].

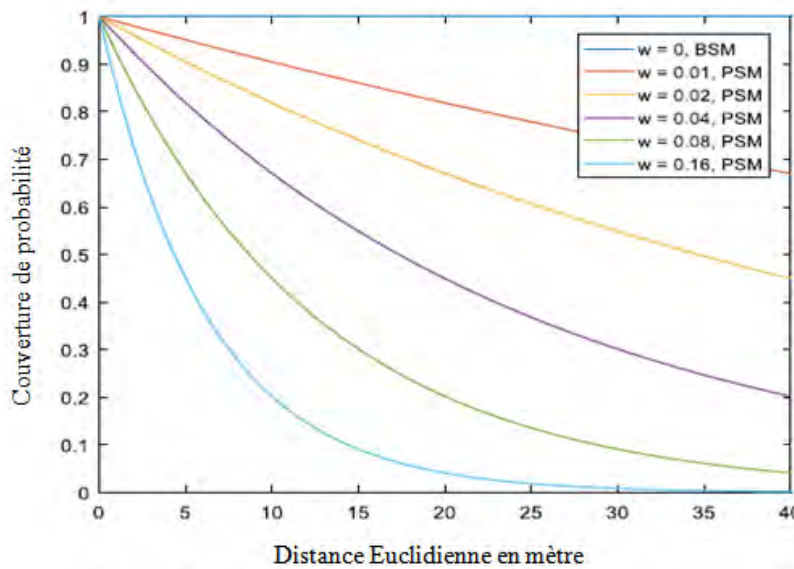
Etant donné que les détections des capteurs sont imprécises, la couverture  $C_{xy}(n_i)$  nécessite donc une incertitude importante elle aussi.

La figure 2.2 représente la probabilité de couverture du point  $p$  ayant des différentes valeurs de  $\omega$  où  $\omega$  présente la plage de détection du capteur. Selon [88] la couverture dans le modèle de détection probabiliste est exprimée en (4).

$$C_{xy}(n_i) = \begin{cases} 0, & \text{si } r_s + r_e \leq d(n_i, p) \\ e^{-\delta a^\beta}, & \text{si } r_s - r_e < d(n_i, p) < r_s + r_e \\ 1, & \text{si } r_s - r_e \geq d(n_i, p) \end{cases} \quad (4)$$

Où  $r_e$  est la mesure d'incertitude dans la détection du capteur.

Et la valeur  $0 < r_e$  et  $r_e < r_s$ ,  $a = d(n_i, p) - (r_s - r_e)$  et  $\delta$  et  $\beta$  sont des paramètres qui mesurent la probabilité de détection dans le cas où une cible se trouve à une certaine distance du capteur et à distance supérieur à  $r_e$ .



**Figure 2. 2 Probabilité de couverture du point p ayant des différentes valeurs de  $\omega$  [15]**

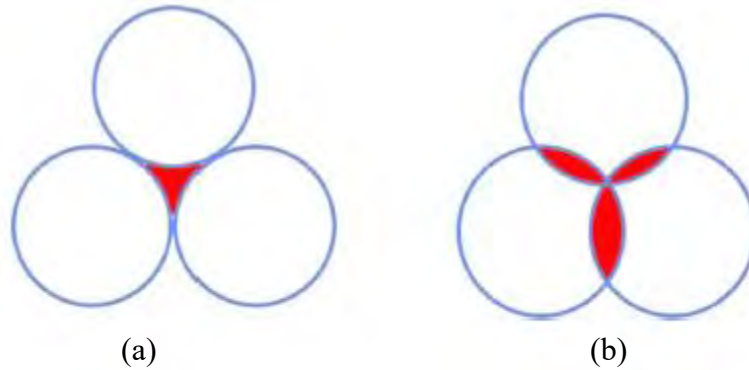
### 2.3. Couverture et connectivité

La couverture d'un réseau de capteurs sans fil fournit la qualité de la surveillance que le réseau peut offrir. Il reste à identifier la qualité de la surveillance d'une région d'intérêt au moyen de capteurs et l'efficacité avec laquelle un réseau de capteurs peut identifier les intrus. Afin de parvenir à une conception et une utilisation efficaces des réseaux de capteurs dans divers scénarios d'application, la couverture d'un réseau de capteurs repose sur de nombreux paramètres [89]. Le problème de couverture peut comporter deux catégories : le problème de 1-couverture et le problème de k-couverture. Pour ce qui est du problèmes de k-couverture, chaque cible dans le champ de détection doit être couverte par au moins k capteurs différents. En ce qui concerne le problème de 1-couverture, le champ de détection doit être couvert au

moins par un capteur. Les auteurs dans [90] ont classifié les stratégies de déploiement de la couverture selon deux catégories principales : la couverture statique et la couverture dynamique.

### 2.3.1. Couverture statique

L'objectif de la couverture statique est d'améliorer les performances de détection globales d'un réseau de capteurs en optimisant les emplacements des nœuds capteurs [91]. Ceci qui peut être considéré comme étant un développement d'un algorithme de distribution de capteurs pour faire en sorte que la zone non-surveillée des capteurs soit la plus faible ou pour minimiser la zone de détection superposée (en rouge) des nœuds de capteur, comme illustré par les figures 2.3 (a) et (b) respectivement. La difficulté de cette problématique est la définition de la portée de détection et son influence sur l'ensemble du réseau. L'objectif est de couvrir entièrement la zone surveillée avec le minimum de nœuds capteurs [90].



**Figure 2. 3 Problème de couverture statique où une partie représente la zone d'exposition en (a) et une partie représente la zone superposée en (b) [90]**

### 2.3.2. Couverture dynamique

La couverture dynamique est définie par l'utilisation d'un réseau de capteurs mobiles et résulte de la mobilité constante des capteurs. Pendant que les capteurs se déplacent, les emplacements qui ont été découverts au début seront potentiellement couverts ultérieurement. Par conséquent, une zone plus large est couverte au fil du temps, et les intrus qui pourraient ne jamais être découverts dans un réseau de capteurs fixes peuvent maintenant être détectés par des capteurs mobiles. D'autre part, cette amélioration de la couverture se fait au détriment du fait qu'une zone n'est couverte qu'une partie de cette période de temps, alternant entre couverte et non couverte. Selon [92], les auteurs ont défini la connectivité réseau comme un concept de la théorie des graphes permettant aux capteurs de communiquer entre eux afin de transmettre leurs données au puits. Un réseau peut être modélisé comme un graphe dans lequel les sommets sont les nœuds capteurs et les arêtes sont le lien de communication entre les nœuds. Dans ce cas, un réseau connecté signifie que le

graphe sous-jacent est connecté. Un **RCD** est dit n-connecté si l'élimination de n'importe quel (n-1) capteurs ne laisse pas le graphe de communication déconnecté [93]. Certaines caractéristiques telles que la qualité, la robustesse et le débit d'un réseau de capteurs sont directement affectées par le degré de couverture et de connectivité [94].

## 2.4. Approches de déploiement dans les RCD

Généralement, le déploiement dans les réseaux sans fil de collecte de données est classifié en deux catégories : le déploiement de capteurs statiques et le déploiement de capteurs dynamiques [95]. Le choix d'une méthode appropriée dépend de divers facteurs, tels que la nature de la zone d'intérêt et le type de capteurs. Après le déploiement, si les capteurs ne changent pas de position, ce type de déploiement est connu sous le nom de déploiement de capteur statique. Lorsque les capteurs peuvent changer de position, ce type de déploiement est appelé déploiement de capteur dynamiques. Il existe deux types de méthodes pour le déploiement de capteurs statiques :

- le déploiement aléatoire,
- le déploiement déterministe.

### 2.4.1. Déploiement aléatoire

Selon [96], le déploiement aléatoire est généralement utilisé dans des zones éloignées ou des environnements contraints. La méthode de déploiement déterministe est optimale si les nœuds de capteurs sont déployés à travers des coordonnées prédéfinies dans un RoI avec la connectivité d'un réseau assurée. Dans certains cas le déploiement déterministe est quasi impossible en raison de l'inaccessibilité ou de la complexité des régions d'intérêt. Dans des situations exceptionnelles telles que le cas de catastrophe naturelle ou de guerre, le déploiement déterministe de capteurs est très peu pratique voire irréalisable. Généralement, dans ce type de conditions, le déploiement aléatoire est la seule option disponible. Dans d'autre cas, le déploiement aléatoire agit comme partie essentielle des autres méthodes de déploiement. Dans le cas de déploiement dynamique de capteurs mobiles, on suppose qu'au départ, le déploiement du capteur se fait d'une façon aléatoire. En déploiement aléatoire, les positions des capteurs sont représenté par une fonction mathématique appelée fonction de densité de probabilité

$$RoI = \{x_1 \geq X \leq x_2, y_1 \geq Y \leq y_2\} \quad (5)$$

Ainsi la fonction de densité de probabilité est assurée par (6) :

$$P(x_1 \geq X \leq x_2, y_1 \geq Y \leq y_2) = \int_{x_1}^{x_2} \int_{y_1}^{y_2} f(x, y) dx dy \quad (6)$$

Selon la stratégie de déploiement, les coordonnées de position des capteurs suivent une distribution spécifique et distincte. [96].

### **2.4.2. Déploiement déterministe**

L'objectif fondamental des méthodes de déploiement déterministes est de pré-calculer la position des nœuds de capteurs au plus tôt pour atteindre les objectifs souhaités. Le déploiement de réseau de capteur sans fil dans le monde réel a lieu selon la méthodologie obtenue. Le concept fondamental est de confirmer que le réseau déployé atteint les objectifs souhaités. À l'heure actuelle, il n'existe pas des méthodes claires et optimales d'un déploiement permettant d'atteindre les objectifs souhaités. De plus, les travaux actuels fournissent des algorithmes de placement sans authentification pratique. Généralement, le déploiement déterministe est utilisé pour les applications intérieures, lorsque les capteurs sont coûteux. Dans une comparaison entre le déploiement déterministe et un déploiement aléatoire, la configuration du réseau optimale est offerte par le déploiement déterministe.

La couverture est considérée comme le facteur le plus important dans le déploiement déterministe. Dans le processus de déploiement, il est préférable d'utiliser un algorithme dans lequel nous trouvons une couverture maximale avec un minimum de nœud capteurs. Les auteurs [15] ont classifié les méthodes de déploiement selon les objectifs recherchés.

### **2.5. Technique de déploiement classique dans les RCD**

Il est question dans cette section de décrire les principales dernières techniques de couverture utilisée dans les réseaux de capteur sans fil.

Il existe diverses applications qui utilisent des techniques de déploiement aléatoire, où les nœuds capteurs sont répartis de manière irrégulière dans le RoI. Certaines régions RoI peuvent avoir un grand nombre de nœuds capteurs, tandis que dans certaines régions, il peut ne pas avoir suffisamment de nœuds capteurs [17]. Ainsi, certains trous de couverture peuvent apparaître, rendant difficile l'obtention d'une couverture complète. La figure 2.4 présente une classification de base des différentes techniques de couverture dans un réseau de capteur sans fil.

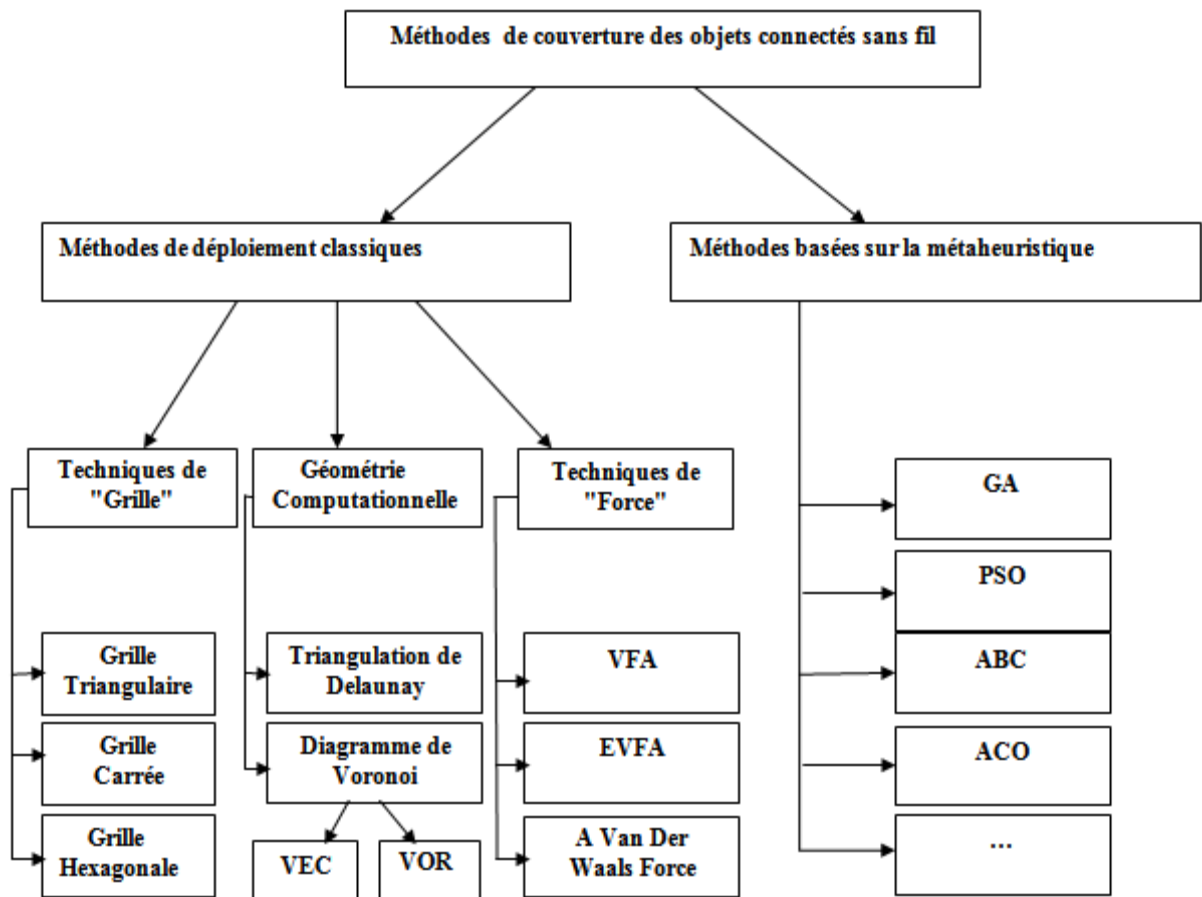


Figure 2. 4 Classification des différentes méthodes de couverture

### 2.5.1. Technique basée sur la force

La technique basée sur la force repose sur une stratégie utilisant l'idée de forces virtuelles, qui peuvent être des forces virtuelles répulsives, attractives ou nulles. Ces forces s'exercent sur une paire de nœuds capteurs adjacents en fonction de distance séparant l'un de l'autre. Si la distance entre deux nœuds capteurs adjacents est supérieure à un seuil prédéfini, alors la force d'attraction agira. De même, si la distance entre deux nœuds adjacents est inférieure à un seuil prédéfini, la force agissante sera de nature répulsive. Si la distance est égale à un seuil donnée, une force nulle sera exercée

#### 2.5.1.1. Algorithme de force virtuelle VFA

Selon [97], dans un algorithme de force virtuelle (VFA), chaque capteur est soumis à trois types de forces.

- Le premier type de force est marqué par les obstacles exerçant une force répulsive ( $F_{iR}$ ) sur les capteurs.
- Le deuxième type de force est caractérisé par la zone de couverture préférentielle exerçant une force attractive ( $F_{iA}$ )

- Le troisième type de force comporte les autres capteurs exerçant une force attractive ou répulsive en fonction de la distance entre les nœuds ( $F_{ij}$ ), comme vu précédemment.

La force résultante finale (force nette) sur un nœud capteur  $n_i$  peut être modélisée par l'équation. (7):

$$F_i = F_{iR} + F_{iA} + \sum_{j+1 \& j \neq i}^K F_{ij} \quad (7)$$

et la force  $F_{ij}$  est exprimée par l'Eq. 8 :

$$\vec{F}_{ij} = \begin{cases} (W_A (d_{ij} - D_{th}), \theta_{ij}) & \text{si } d_{ij} > D_{th} \\ 0 & \text{si } d_{ij} = 0 \\ (W_R \frac{1}{d_{ij}}, \theta_{ij} + \pi) & \text{si } d_{ij} < D_{th} \end{cases} \quad (8)$$

- $d_{ij}$  représente la distance qui sépare entre le nœud  $n_i$  et le nœud  $n_j$
- $\theta_{ij}$  est l'orientation entre le nœud  $n_i$  et le nœud  $n_j$
- $D_{th}$  représente la valeur seuil de la distance.

Par ailleurs, VFA est une approche centralisée dans laquelle le puits agit comme un nœud capteur de prise de décision qui calcule également les nouvelles positions et la direction du mouvement pour chaque nœud capteur. Selon [97,98], VFA fournit un bon degré de couverture pour surveiller le RoI et maintenir la connectivité du réseau. Il fournit également une bonne approche pour gérer les obstacles présents dans le RoI. Mais en raison de sa fonctionnalité complexe, il existe plusieurs limites car seuls les nœuds mobiles peuvent être utilisés; de plus et le nœud récepteur doit avoir une grande capacité de calcul.

### **2.5.1.2. Algorithme de force virtuelle distribuée**

Dans l'algorithme de force virtuelle distribuée (DVFA) [99], il s'agit des nœuds capteurs mobiles redéployés qui ont été initialement déployés de manière aléatoire pour s'assurer du résultat souhaité en termes de degré de couverture et de connectivité tout en obtenant une faible consommation d'énergie par nœud. Aucune synchronisation n'est requise entre les nœuds capteurs et ils peuvent se déplacer en fonction de la force exercée sur eux par leurs nœuds voisins.

### **2.5.1.3. Déploiement basé sur la force de Van der Waals**

Il existe plusieurs types de forces virtuelles pouvant agir sur un nœud capteur telles que la force agissant sur un nœud en raison de ses nœuds adjacents, la force de frottement (s'opposant au mouvement du nœud) et la force employée par l'événement. Pour plus de commodité, si nous ne considérons pas les effets



environnementaux sur les nœuds, la force nette sur n'importe quel nœud peut être représentée par l'équation (9)

$$F_i(t) = F_i^e + F_i^f \quad (9)$$

Dans le cas de  $F_i^e(t)$  il est question d'un processus d'échange entre les nœuds et la force de frottement qui est représentée par  $F_i^f(t)$ . La force d'échange est supposée désigner les forces de Van der Waal [100]. En physique moléculaire, les forces de Van der Waals sont des interactions dépendant de la distance entre des molécules ou des atomes. Cette force qui ne provient pas d'une liaison chimique est de nature très faible. Van der Waal représente la force d'interaction entre deux nœuds adjacents, il est possible d'utiliser la Triangulation de Delaunay pour obtenir une relation adjacente [101].

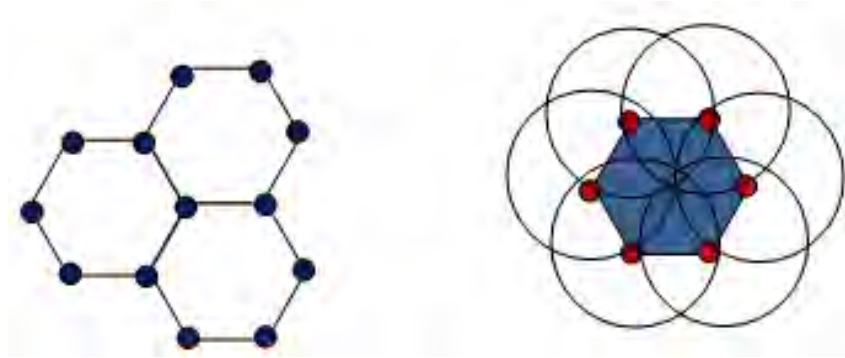
### 2.5.2. Technique basée sur la grille

Dans la technique basée sur la grille [102], les sites des nœuds capteurs sont déterminés sur la base d'un motif de grille, qui peut avoir une forme rectangulaire, une forme triangulaire ou une forme hexagonale. Les nœuds sont placés au centre de la grille et la zone de surveillance est divisée en petits morceaux. Les auteurs [103] proposent une solution pour trouver un déploiement optimal afin d'avoir une p-couverture et une q-connectivité pour certaines valeurs typiques de  $r_c$  et  $r_s$ , comme déjà mentionné précédemment. Pour établir des différents rapports entre  $r_c$  et  $r_s$ , ils ont introduit la grille carrée, la grille triangulaire et la grille hexagonale. Les auteurs de cette étude ont comparé le nombre de nœuds capteurs nécessaires pour les modèles de déploiement afin d'obtenir une couverture p et une connectivité q avec  $r_s = 30$  m à l'aide de l'équation suivante :

$$\text{Nombre des nœuds} = \frac{\text{Zone cible}}{Ta, Sa \text{ ou } Ha} \quad (10)$$

Où  $Ta$  est l'aire triangulaire,  $Sa$  est l'aire carrée et  $Ha$  est l'aire hexagonale.

Un algorithme de couverture basé sur une grille carrée virtuelle a été introduit dans [104], chaque nœud à sa région de détection divisée sous la forme d'une grille carrée virtuelle. Un nœud cible est dit redondant si des grilles complètes ont été également couvertes par les nœuds voisins. En outre, il assure une couverture complète garantie dans le RoI. Cependant, cet algorithme ne prend pas en compte le scénario d'obstacles ou la mobilité des nœuds.



**Figure 2. 5 Grille hexagonale**

Le tableau 2.1 montre les techniques de grille et de la force virtuelle, en discutant parallèlement de leurs avantages et de leurs inconvénients.

**Tableau 2. 1 Comparaison entre les différentes techniques basées sur la grille et sur la force virtuelle**

Techniques	Avantages	Inconvénients	Complexité
<b>Technique basée sur la force</b>			
Van der Waals [100]	- Taux de couverture élevé - Temps de convergence idéale en présence des obstacles	- La connectivité et la consommation d'énergie ne sont pas prises en considération	élevée
VFA [97]	- Fournir une couverture adéquate pour les zones de surveillance tout en maintenant la connectivité réseau - Prendre en considération les obstacles	- Utiliser seulement pour les nœuds mobiles - Une capacité de calcul très élevés pour les puits. - Performances indésirables pour les réseaux hétérogènes en raison des forces virtuelles	élevée
DVFA [99]	Assurer la couverture et la connectivité La consommation d'énergie est prise en considération	Coût élevé en termes de distance parcourue et en termes de temps	Moyenne
<b>Technique basée sur la grille</b>			
Triangulaire [106]	Une couverture complète avec minimisation de nombre de nœuds à déployer	La connectivité n'est pas considérée	élevée
Carré [107]	Une couverture et une connectivité avec un nombre réduit de nœuds Réduire la consommation d'énergie Moins de complexité de calcul par rapport au technique triangulaire (utilisation de l'algorithme de planification de sommeil SSA)	Les obstacles ne sont pas pris en compte Les nœuds mobiles ne sont pas applicables	Moyenne
Triangulaire carré et hexagonale [108]	Fournir une couverture multiple par rapport aux techniques triangulaire et carré. Assurer la connectivité du réseau	La couverture varie selon le ratio $r_c$ et $r_s$	Moyenne

### 2.5.3. Techniques basées sur la géométrie computationnelle

La stratégie de géométrie computationnelle [109] est basée sur des objets géométriques tels que des points, des polygones et des segments de ligne. La plupart des techniques basées sur des modèles irréguliers sont plus complexes que les techniques classiques. Le Diagramme de Voronoï (DV) et la Triangulation de Delaunay (TD) [110] sont les méthodes de géométrie computationnelle les plus populaires qui sont utilisées dans les réseaux de capteurs sans fil.

#### 2.5.3.1. Le Diagramme de Voronoï DV

Le diagramme de Voronoï est une méthode de partitionnement de la zone d'intérêt en un certain nombre de polygones de Voronoï en fonction des distances entre les nœuds capteurs. Chaque nœud occupe seulement un polygone, où tous les points intérieurs de ce polygone sont plus proches de tous les autres capteurs. La figure 2.6 montre un Diagramme de Voronoï.

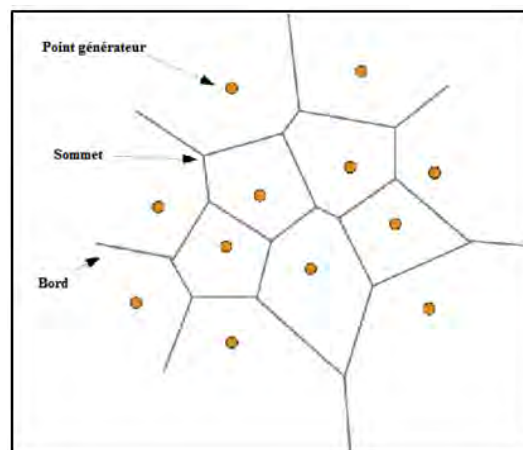


Figure 2. 6 Diagramme de Voronoï

Sung et Yang [111] ont introduit le Diagramme de Voronoï et l'algorithme glouton distribué pour déployer les capteurs sans fil directionnels (capteurs ayant une direction de détection limitée). Leur algorithme proposé est valable pour des capteurs hétérogènes qui ont une plage de détection différente et des angles de détection différents. Les auteurs de cette étude ont amélioré la couverture pour la rotation des capteurs directionnels.

L'algorithme à base vectorielle (VEC) et l'algorithme à base de Voronoï (VOR) ainsi que l'algorithme à base Mini-Max sont des algorithmes bien connus utilisés pour détecter les problèmes de trou de couverture. L'algorithme VEC est tiré du comportement des particules électromagnétiques et éloigne les nœuds capteurs de la zone couverte surpeuplée. Deux capteurs exercent une force répulsive lorsque la distance séparant deux nœuds est faible. Pour ce qui est de l'algorithme VOR, celui-ci retire les nœuds capteurs vers la zone peu couverte. Enfin, l'algorithme Mini-Max est similaire au VOR, car il réduit les trous de couverture en rapprochant les nœuds

capteurs du sommet de Voronoi le plus éloigné. Mini-Max choisit la position du nœud cible comme le point à l'intérieur du polygone de Voronoi qui minimise la distance au sommet de Voronoi le plus éloigné.

Le principal avantage commun de VEC, VOR et Mini-Max consiste à mettre en œuvre des algorithmes distribués qui ont la capacité de s'étendre à un déploiement important en raison de leur capacité de réduire les trous de couverture, ce qui permet de garantir la connectivité. Ainsi ces algorithmes semblent avoir la capacité de faire face aux obstacles dans la zone de surveillance comme l'algorithme BUG2 [112] où le capteur suit la limite de l'obstacle en utilisant la règle de la main droite.

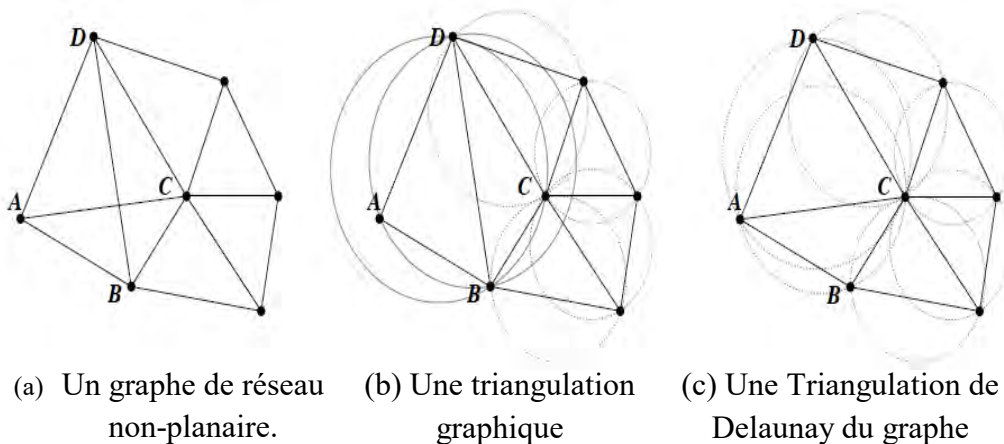
Dans le contexte de déploiement des nœuds dans un réseau de collecte de données, les approches basées sur le Voronoï ont souvent pour but de maximiser la couverture à travers la détection des trous de couverture [113,114].

### 2.5.3.2. Triangulation de Delaunay DT

La Triangulation de Delaunay est la structure duale de Diagramme de Voronoi dans le plan 2D. TD se base sur la propriété de l'ensemble des cercles vides, c'est-à-dire que pour chaque arête de la Triangulation de Delaunay, il est possible de trouver un cercle passant par les extrémités du bord sans enfermer d'autres points [115].

En général, la triangulation est une subdivision d'un objet géométrique en simplexes. La triangulation d'un ensemble discret de points est une subdivision de l'enveloppe convexe des points en simplexes de telle sorte que deux simplexes quelconques se recoupent en une seule face commune et que les sommets des simplexes de subdivision coïncident avec les points [116].

Plusieurs algorithmes ont été proposés pour la triangulation dans les réseaux de capteurs sans fil [117, 118,119]. Un graphe de réseau est généralement non-planaire (comme le montre les arêtes qui se croisent sur la figure 2.7 (a)), alors qu'une triangulation est un graphe planaire, où aucune arête ne se croise (comme illustrée sur la figure 2.7 (b)).



**Figure 2. 7 Triangulation et Triangulation de Delaunay d'un graphe de réseau non-planaire [120]**

La triangulation de Delaunay est une triangulation spécifique définie comme suit.

***Définition.** Une triangulation de Delaunay d'un ensemble de points sur un plan, est une triangulation où aucun point ne peut exister à l'intérieur du cercle circonscrit d'un triangle [121].*

Un exemple de Triangulation de Delaunay est illustré par la figure 2.7 (c), où aucun des cercles circonscrits aux triangles ne contient de nœud. D'autre part, la triangulation de la figure 2.7 (b) ne répond pas à la définition ci-dessus. Dans ce cas le cercle circonscrit de triangle ABD contient le nœud C. Il ne s'agit donc pas d'une Triangulation de Delaunay. La Triangulation de Delaunay est préférable dans des nombreux algorithmes de réseau, car elle a tendance à éviter les triangles maigres en maximisant l'angle minimum dans la triangulation. De plus, une meilleure localisation peut être obtenue sur la base d'une Triangulation de Delaunay que celle qui est basée sur une Triangulation non-Delaunay [122,123]. La Triangulation de Delaunay avec toutes les arêtes égales s'appelle une Triangulation de Delaunay équilatérale.

## **2.6. Les approches d'optimisation dans le déploiement de réseaux sans fil de collecte des données**

Un algorithme d'optimisation peut déplacer des nœuds capteurs afin de combler les trous. En général, les approches existantes sont souvent utilisées afin de placer d'une manière optimale les capteurs. Ceci revient à dire que les approches sont souvent classées en méthodes locales et globales. Dans cette partie, nous introduisons le concept d'optimisation et ses catégories générales, puis nous présentons les méthodes de déploiement dans un RCD basées sur la méta-heuristique.

### **2.6.1. Concept général d'optimisation et ses catégories**

L'optimisation est le processus d'ajustement des entrées ou des caractéristiques d'un appareil, basé sur une démarche mathématique ou sur une expérience dans le but de trouver la sortie ou le résultat maximum ou minimum. L'entrée inclut les variables, le processus ou la fonction, cette fonction est appelée la fonction de coût, fonction objectif ou fonction de fitness ainsi le résultat représente le coût ou la valeur adaptative [124].

Selon [125], les algorithmes d'optimisation peuvent être divisés en six catégories comme le montre la figure 2.8

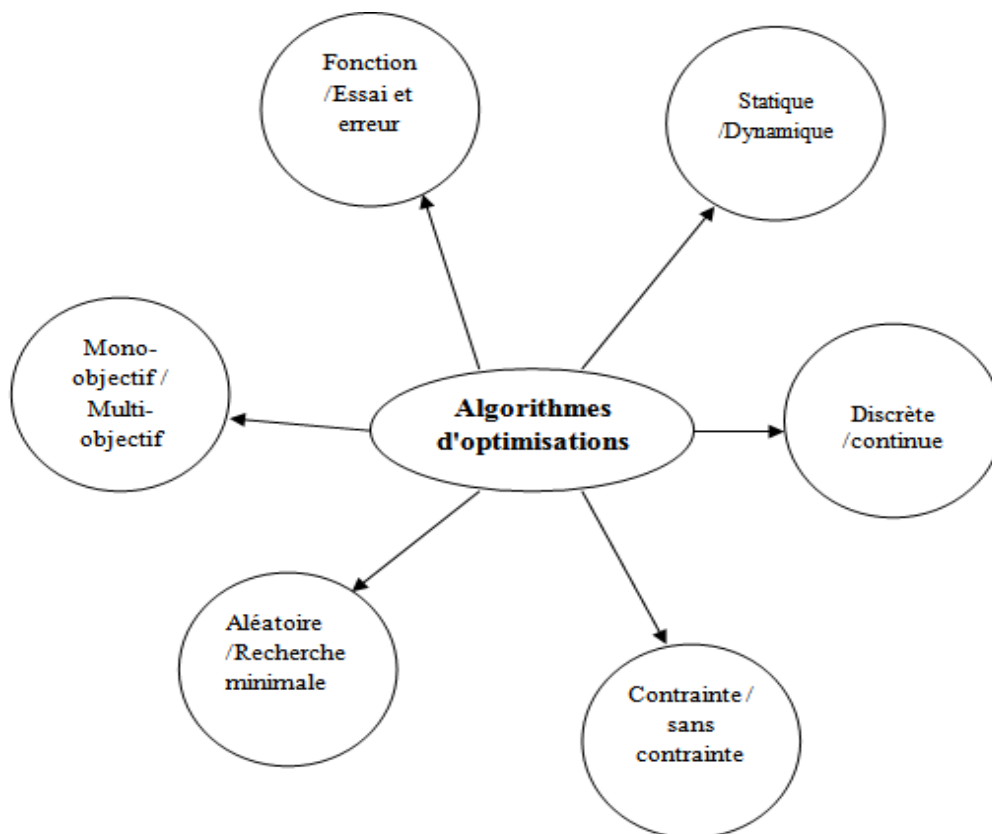


Figure 2. 8 Catégories d'algorithmes d'optimisation

- **L'essai et l'erreur** signifie que le processus d'optimisation trouve les sorties sans connaître des données précises sur le processus de production des résultats. En cas d'optimisation de fonction, une formule mathématique est utilisée pour décrire la fonction objective.
- **L'optimisation dynamique** fait référence à un processus d'optimisation dans lequel la sortie est fonction du temps, alors que dans **l'optimisation statique**, la sortie est indépendante du temps. L'ajout du temps dans l'optimisation dynamique augmente la difficulté de résolution du problème, car la solution évolue dans le temps. Dans le cas de l'optimisation dynamique où l'on recherche l'itinéraire le plus rapide, l'itinéraire le plus court n'est pas nécessairement le plus rapide en raison des dépendances au trafic, des limites de vitesse, des accidents, etc.
- Les algorithmes d'optimisation diffèrent en fonction de type de variables utilisés, ces variables peuvent être **discrètes ou continues**. Les variables discrètes ont un nombre fini de valeurs possibles, alors que les variables continues ont un nombre infini de valeurs possibles. Les applications peuvent être différentes : la commande d'une série de tâches sur une liste est effectuée à l'aide d'une optimisation discrète, tandis que l'optimisation continue est utilisée pour trouver le minimum d'une fonction sur une série de nombres réels.

- Dans la plupart des problèmes d'optimisation, il y a toujours des restrictions imposées par les caractéristiques spécifiques de l'environnement ou des ressources disponibles (telles que des limitations physiques, des restrictions de temps, etc.). Ces restrictions doivent être satisfaites pour qu'une solution soit considérée comme acceptable. Toutes ces restrictions sont en général appelées **contraintes** et décrivent les dépendances entre les variables de décision et les constantes (ou paramètres) impliquées dans le problème. Dans l'optimisation **sans contraintes**, toutes les valeurs peuvent être prises par les variables, tandis que dans l'optimisation en présence de contraintes, les égalités et les inégalités des variables sont incorporées dans la fonction du coût.

- Les algorithmes d'optimisation convexes minimisent les fonctions convexes sur des ensembles convexes en partant d'un ensemble initial de valeurs variables en utilisant les méthodes de calcul traditionnels. En mathématiques, une fonction est dite convexe si le segment de ligne entre deux points quelconques sur le graphique de la fonction se situe au-dessus du graphique, dans un espace euclidien ayant du moins deux dimensions. L'avantage de l'utilisation des algorithmes d'optimisation est considéré comme étant convexe si **un minimum local** existe, dans ce cas il s'agit d'un minimum global. L'optimisation convexe est marquée par une autre caractéristique qui exige que chaque fonction soit strictement convexe, si la fonction a un minimum, alors le minimum est unique. Les algorithmes **aléatoires** (non convexes) utilisent des calculs probabilistes pour trouver les résultats. Ces algorithmes peuvent être utilisés sur des fonctions qui ne sont pas continues ou différentiables. Ces méthodes sont également connues sous le nom de méthodes de recherche directe.

- L'optimisation est **mono-objective** si une seule variable est supposée être optimisée. L'optimisation multi-objective vise à optimiser simultanément plus d'une variable par rapport à un ensemble de contraintes. L'augmentation du nombre de dimensions d'optimisation rend le processus d'optimisation de plus en plus difficile. Dans certains cas, l'optimisation d'un objectif conduit à l'optimisation des autres, ce qui ne doit pas être considéré comme un problème d'optimisation multi-objectifs.

### **2.6.2. Méthodes de déploiement dans RCD basées sur la méta-heuristique**

Des nombreux algorithmes d'optimisation stochastique ont été conçus pour résoudre différents problèmes d'optimisation. On peut citer, parmi ces algorithmes couramment utilisés dans le déploiement de réseaux de collecte des données : l'algorithme génétique (**GA**) [126,127], l'optimisation des essaims de particules (**PSO**) [128,129], la colonie d'abeilles artificielles (**ABC**) [130], l'optimisation des colonies de fourmis (**ACO**) [131], l'optimisation des mouches des fruits (**FOA**) [132], l'algorithme d'optimisation de la recherche de nourriture bactérienne (**BFO**)



[133], l'optimiseur de recherche de groupe (GSO) [134], la recherche par l'algorithme d'harmonie (HSA) [135], la recherche de système chargé (CSS) [136], la dynamique de formation des rivières (RFA) [137]. De plus, différentes combinaisons et hybridations entre ces algorithmes ou avec d'autres paradigmes sont parfois proposées pour améliorer les performances du déploiement en termes de taux de convergence et de précision de la solution.

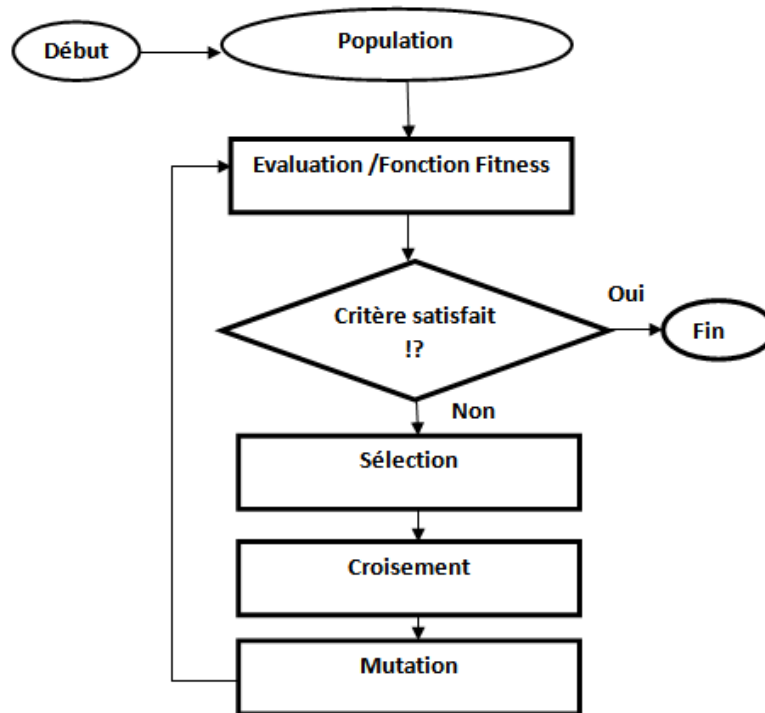
Une branche des méta-heuristiques est appelée algorithme basé sur la population (PBA) tel que l'algorithme AG et l'algorithme ABC. La capacité de recherche de PBA provient non seulement de sa capacité à rechercher dans plusieurs directions à la fois pour trouver la solution possible, mais aussi de sa capacité d'échanger les informations de recherche entre les solutions. Par conséquent, PBA peut généralement trouver la solution approximative très rapidement.

Nous nous intéressons dans cette partie aux algorithmes d'optimisations utilisées dans notre propre déploiement tel que l'algorithme génétique et l'algorithme de Colonie des abeilles artificiels

### **2.6.2.1. L'Algorithme Génétique AG**

L'AG est l'algorithme évolutionnaire méta-heuristique, il s'agit d'une technique de recherche utilisée principalement pour l'optimisation aléatoire qui offre la solution la plus optimale parmi les solutions disponibles [138,139].

Le processus de l'AG commence par un "ensemble des individus", ou appelés encore "solutions possibles", générées aléatoirement appelées également "population". Chaque solution est représentée sous la forme d'une chaîne ou d'un tableau de gènes. Une fonction de fitness sera attribuée pour chaque individu. Plus la valeur de fitness est élevée, plus la solution est proche de la meilleure solution. Par la suite on applique la phase de croisement où deux solutions (ou parents) seront sélectionnés au hasard, et ceci dans le but de produire deux nouvelles solutions (ou enfants). Ensuite la phase de mutation est appliquée sur des chromosomes choisis aléatoirement. Les solutions de la génération suivante sont sélectionnées dès que les processus de croisement et de mutation se sont produits. Ce processus de l'AG est répété jusqu'à l'algorithme soit satisfait. Les étapes d'un AG standard sont décrites dans la figure 2.9.



**Figure 2. 9 Organigramme de l'algorithme Génétique**

L'algorithme génétique [140, 141] est certainement le PBA le plus connu car il a été appliqué à des divers domaines ou problématiques. Plusieurs études [142,143, 144, 145, 146] ont opté pour l'utilisation de l'AG dans le but de traiter le problème de déploiement. Les mécanismes, les opérateurs et les stratégies efficaces d'AG pour différents problèmes d'optimisation sont utilisés dans ces études pour trouver de meilleures solutions que d'autres algorithmes de déploiement, tels que la sélection de tournois. La caractéristique de [147] est que la question de la région d'intérêt (ROI) a été adoptée par AG pour résoudre le problème de déploiement en présence des obstacles et pour minimiser le nombre de capteurs. Les résultats expérimentaux de [147] montrent que AG surpasse les algorithmes MAX\_AVG\_COV et MAX\_MIN\_COV. Plusieurs combinaisons de différents opérateurs d'AG ont été testées dans [148] pour résoudre le problème de déploiement. La stratégie aléatoire de sélection des chromosomes avec croisement en deux points et mutation de coordonnées aléatoires peut trouver un meilleur résultat que les autres combinaisons. Une étude ultérieure [145] considère également plus d'un facteur pour le problème de déploiement afin de maximiser la zone de couverture de détection tout en minimisant l'énergie. Dans cette recherche, la distance de déplacement des capteurs peut être considérée comme similaire à l'énergie consommée, car la distance entre les capteurs implique un coût d'énergie de transmission consommée. Par contre, des études [142, 143, 144] ont accordé plus d'importance aux facteurs pour la résolution de problème de déploiement, alors que d'autres études ont opté pour des modifications dans plus d'un opérateur dans le but d'améliorer les performances d'AG.

Dans [143], le chromosome est conçu pour fournir une structure hybride qui contient la coordination des capteurs et leur état de densité. De plus, la capacité d'AG de résoudre le problème de déploiement peut également être trouvées dans des études pertinentes. Par exemple, AG est utilisé pour trouver la solution pour le déploiement des agents mobiles [149] afin de réduire la consommation d'énergie des agents mobiles sur les capteurs. Dans [150], AG est utilisé pour savoir comment déployer les capteurs mobiles pour réduire les trous de couverture à travers la modification de la fonction fitness, l'opérateur de croisement, l'opérateur de mutation et la stratégie de sélection [147] pour améliorer les performances dans la résolution de problème de déploiement. Contrairement aux études utilisant la roulette ou le tournoi, la sélection a été utilisée pour augmenter la probabilité de conserver les bons chromosomes aux générations futures.

### **2.6.2.2. L'algorithme de Colonie des Abeilles Artificielles ABC**

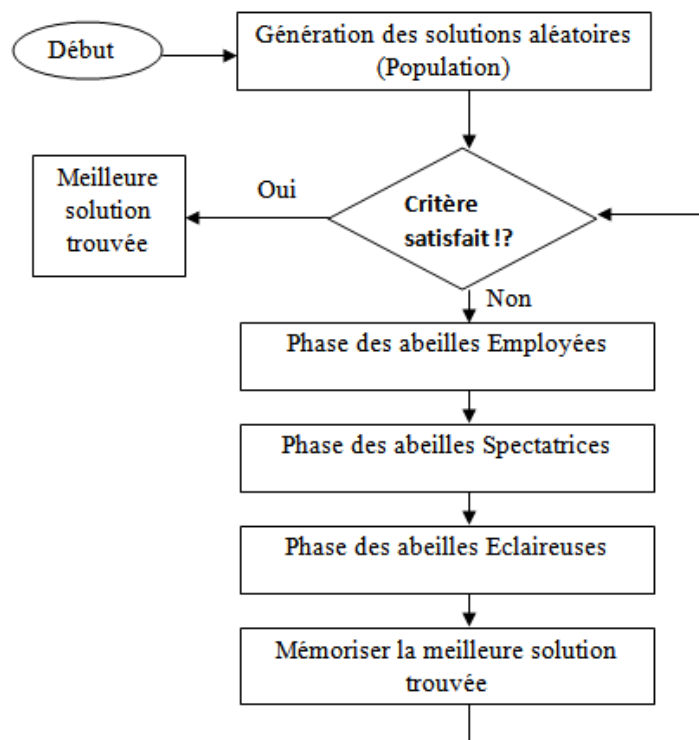
La colonie d'abeilles artificielles [151] est une technique d'optimisation méta-heuristique, inspirée par le comportement de butinage d'un essaim d'abeilles mellifères. Il résout la question de l'optimisation continue de manière répétitive.

Une colonie d'abeilles peut s'étendre sur de longues distances afin d'utiliser simultanément un grand nombre de sources de nourriture [152]. Les abeilles sont divisées en trois catégories, les abeilles employées, les spectatrices et les éclaireuses. Au début toutes les abeilles utilisant la source de nourriture sont des abeilles employées. Ces abeilles profitent de la source de nourriture, et elles rapportent des informations sur la source de nourriture à la ruche et partagent par la suite ces informations avec les abeilles spectatrices.

Les abeilles spectatrices dans la ruche attendent que les abeilles employées partagent des informations sur les sources de nourriture qu'elles ont découvertes, tandis que les abeilles éclaireuses seront toujours à la recherche de nouvelles sources de nourriture près de la ruche. Les parcelles alimentaires qui peuvent facilement fournir de grandes quantités de nectar ont tendance à être visitées par plus d'abeilles, tandis que les parcelles avec moins de nectar reçoivent moins d'abeilles [153]. Les abeilles employées partagent des informations sur la source de nourriture en dansant dans des zones désignées au sein de la ruche. La nature de la danse est proportionnelle à la teneur en nectar de la source de nourriture qui vient d'être exploitée par les abeilles dansantes. Les abeilles spectatrices regardent la danse et choisissent une source de nourriture en fonction d'une probabilité proportionnelle à la qualité de la source de nourriture. Par conséquent, les bonnes sources de nourriture attirent plus de spectateurs que les mauvaises. Chaque fois qu'une source de nourriture est pleinement utilisée, toutes les abeilles associées à cette zone abandonnent cette source de nourriture et deviennent des éclaireuses. Le rôle des abeilles éclaireuses peut être considéré comme un travail d'exploration et les abeilles spectatrices peuvent être imaginés en train de faire des travaux d'exploitation. Le processus de

recherche de nourriture commence par une colonie. Les abeilles éclaireuses sont envoyées pour trouver des parcelles de nourriture prometteuses.

Lors de retour à la ruche, les abeilles éclaireuses qui trouvent des zones au-dessus d'un certain seuil de qualité se dirigeront vers la piste pour exécuter une danse. Cette danse est essentielle à la communication de la colonie et contient des informations sur les zones alimentaires car elle présente la direction dans laquelle elle se trouvera, sa distance à la ruche et son indice de qualité (ou fitness) [152,153]. Cette information aide la colonie à diriger ses abeilles précisément vers la zone de nourriture. Les étapes d'un ABC standard sont décrites dans la figure 2.10.



**Figure 2. 10 Organigramme de l'algorithme ABC**

Les algorithmes basés sur les colonies d'abeilles sont étudiés par Karaboga et al. [155]. Une comparaison d'ABC avec l'algorithme de rétro-propagation traditionnel et l'algorithme génétique réalisé par [156] montre que l'algorithme ABC surpasse les deux autres et peut être utilisé pour former des réseaux de neurones à propagation avant (*feed-forward*). Dans ce réseau, l'information ne se déplace que dans une seule direction, vers l'avant, à partir des nœuds d'entrée, en passant par les couches cachées (le cas échéant) et vers les nœuds de sortie.

Une autre comparaison d'ABC avec les algorithmes d'évolution différentielle (DE) et d'optimisation d'essaim de particules (PSO), par [157], montre qu'ABC est plus

performant et peut être utilisé pour résoudre les problèmes d'ingénierie multimodaux à haute dimensionnalité.

Dans le contexte du déploiement dans les réseaux de collecte des données, les auteurs de [158] ont proposé une optimisation d'ABC pour un déploiement dynamique. Cet algorithme donne de meilleures performances tout en augmentant la zone de couverture. Par contre, cet algorithme ne considère pas la connectivité à la station de base.

Les auteurs [159] ont proposé une méthode basée sur l'algorithme ABC pour résoudre le problème de couverture de surface pour un espace irrégulier bidimensionnel. La zone considérée est une surface irrégulière à deux dimensions. L'objectif est de trouver les positions des nœuds capteurs afin de minimiser la portée de détection pour un problème de couverture simple.

### **2.6.2.3. Autres algorithmes**

**L'optimisation des essaims de particules PSO** développée par Eberhart et Kennedy [160] est une technique d'optimisation puissante. Cette technique s'inspire du comportement des groupes d'oiseaux. Les auteurs de [161] ont utilisé la technique d'optimisation par essaim de particules pour optimiser le problème de couverture du réseau dans les réseaux de capteurs sans fil. Les auteurs ont utilisé un modèle de détection de probabilité, un déploiement optimal a été trouvé par PSO pour une recherche globale. L'efficacité de ce protocole a été démontrée avec des résultats de simulation dans MATLAB.

**L'optimisation des colonies de fourmis ACO** est une technique basée sur les probabilités pour répondre aux défis d'optimisation en réduisant le problème en un problème de recherche de chemin par un graphe. Les fourmis ont l'habitude de trouver de la nourriture près de leur nid de manière aléatoire. Lorsque l'une des fourmis trouve la nourriture, elle analyse la qualité et la quantité de la nourriture et par conséquent, elle établit une piste chimique. Ces pistes chimiques aident les fourmis à rechercher le chemin le plus court entre le nid et la nourriture [162]. L'auteur de [163] a appliqué l'optimisation des colonies de fourmis pour résoudre le problème de redéploiement des nœuds capteurs mobile déployé de manière aléatoire, il a conclu que cet algorithme consomme un minimum d'énergie mais provoque un retard par rapport au GPSR (Greedy Perimeter Stateless Routing), ce dernier est un protocole de routage géographique efficace et éprouvé pour un réseau ad hoc sans fil, il offre également une prise en charge du routage pour le réseau de capteurs sans fil. L'ACO assure une mise en œuvre dynamique du réseau tout en considérant raisonnablement tous les paramètres significatifs pour obtenir un chemin optimal parmi tous les chemins les plus courts. ACO garantit également la convergence du réseau mais ne considère pas l'obstacle.

**Le recuit simulé SA** [164] est une approche méta-heuristique conçue pour découvrir l'optimum global pour d'une fonction donnée. Le recuit est un processus dans lequel un métal est chauffé au-dessus de son point de fusion, puis cette température est maintenue, ensuite il est refroidi très progressivement pour donner au métal une structure cristalline parfaite. Dans le contexte de déploiement optimal d'un réseau de capteur sans fil, les auteurs [165] ont proposé un algorithme hybride. Il résout un problème pour des utilisations de "1 couverture" et de "1 connectivité". Cette approche est un hybride de la méthode du gradient et de l'algorithme SA et se concentre sur l'utilisation d'un nombre inférieur de nœuds. Cette approche a pu optimiser la position des nœuds et le nombre de nœuds capteurs, pour obtenir une couverture maximale dans un champ d'intérêt.

L'algorithme Krill Herd (**KH**) [166] est fondé sur l'imitation des actions de rassemblement des créatures de Krill. Dans cet algorithme, l'objectif est d'optimiser les plus petites distances entre chaque krill de sa nourriture et de la plus grande concentration du groupe pour le mouvement du krill. L'emplacement de l'individu krill peut être assemblé par trois éléments, un mouvement influencé par l'existence de différents krills, en cherchant l'action et une diffusion aléatoire. Il existe deux schémas globaux et deux schémas locaux exécutés simultanément, ce qui améliore considérablement l'efficacité de l'algorithme KH. KH a montré des meilleurs résultats que plusieurs autres méta-heuristiques.

**Une version unique du groupe de krill** a été proposée par les auteurs de [167] nommée algorithme du groupe de krill (**SKH**). Dans ce schéma, une stratégie de reproduction génétique modifiée appelée opérateur de sélection et de croisement des étalons (**SSC**) a été introduite dans KH au cours du processus de mise à jour du krill pour résoudre divers problèmes d'optimisation numérique. L'opérateur SSC, composé d'une sélection et d'une opération de croisement, est utilisé pour trouver une bonne solution. Il fournit un résultat plus précis que le KH. Sa limite est pour affiner les paramètres pour chaque problème unique de la vie réelle.

L'algorithme de déploiement basé sur **la recherche d'harmonie (HS)** est proposé par les auteurs [168] qui ont utilisé l'algorithme HS pour maximiser la couverture d'un réseau avec un coût minimum en utilisant un déploiement déterministe. Un codage de longueur variable est utilisé pour chaque solution candidate pour représenter un nombre variable de nœuds de capteurs dans chaque solution. La dérivation d'une fonction multi-objectif est fournie. Il a pris en compte le taux de couverture, le nombre de nœuds déployés et la distance minimale entre les nœuds. Les résultats expérimentaux ont prouvé la capacité de leur algorithme à atteindre une couverture complète avec un nombre minimum de nœuds. De plus, il a eu rapidement de convergence par rapport à un algorithme génétique. Cependant, les

auteurs n'ont pas pris en compte la connectivité du réseau et la consommation d'énergie des nœuds.

### **2.6.3. Hybridation entre DV et des algorithmes d'optimisation**

Les heuristiques les plus connus pour le déploiement et l'optimisation dans les réseaux de collecte de données IoT résident dans l'optimisation par l'algorithme génétiques AG qui a pour objectif principal d'améliorer les positions des nœuds afin de maximiser la couverture d'un champ d'intérêt et de prolonger par la suite la durée de vie du réseau. L'AG a aussi pour objectifs importance la réduction de la consommation d'énergie, l'optimisation par l'essaim de particules (PSO), l'optimisation par les colonies de fourmis (ACO) qui est utilisé généralement pour optimiser les chemins de routage entre les nœuds. On remarque également les approches basées sur les Voronoï qui sont utilisés dans la plupart des travaux pour maximiser la couverture dans un champ d'intérêt par la détection des trous de couverture dans les réseaux de collecte de données IoT.

En conclusion, nous avons mis l'accent dans cette section sur les études suggérant une hybridation entre DV et des algorithmes d'optimisation pour différentes applications et contextes. Ils existe des travaux qui ont été réalisés sur la combinaison des approches basées sur Voronoï avec d'autres algorithmes inspirés de la nature tels que l'algorithme d'optimisation par les colonies de fourmis ACO, l'optimisation par l'essaim de particules PSO et les algorithmes génétiques AG comme le montre le tableau 2.2.

**Tableau 2. 2 Hybridation de Voronoi avec des algorithmes d'optimisations  
ACO, PSO et GA**

	Méthode d'hybridation	Fonctionnement	Impact
Voronoi + ACO [131,169,170]	Voronoi: déterminer tous les chemins possibles dans un réseau	Attribuer des valeurs de poids aux Voronoi edges pour la recherche de chemin de routage.	-La distribution des nœuds -Les valeurs de poids - La fonction d'évaluation.
	ACO: identifier le chemin le plus court parmi tous les chemins générés par Voronoi		
Voronoi + PSO [128, 171,172]	Voronoi: détecter les trous de couverture dans un champ d'intérêt	Générer des points virtuels à partir des trous détectés. La plage de détection parfois modifiée.	-Les points virtuels -L'emplacement des nœuds -La vitesse des nœuds -La meilleure solution locale ou globale.
	PSO: générer des Voronoi vertices (des points virtuels) ou des points limites aléatoires pour réduire la consommation d'énergie et pour maximiser la durée de vie du réseau.		
Voronoi + GA [173, 174, 175, 178]	Voronoi: détecter les trous de couverture dans un champ d'intérêt	Détection des trous puis repositionnement des nœuds à déployer, modification au niveau de distribution des nœuds	-Impact sur la fonction d'évaluation. -Impact sur le pourcentage de couverture des trous -L'ajout parfois des nœuds mobiles supplémentaires aux trous détectés
	GA: générer des nouvelles solutions candidates c'est à dire les nouveaux emplacements des nœuds dont le but de maximiser la couverture d'un champ d'intérêt.		

Il existe beaucoup des travaux sur des hybridations entre Voronoi et l'optimisation des colonies de fourmis ACO pour le déploiement des nœuds dans un réseau. Principalement l'option de ces travaux pour une l'utilisation de DV a été conçu pour optimiser la planification du chemin pour une communication efficace entre les



nœuds du réseau. Le premier rôle d'ACO est de trouver les chemins de connexion dans un réseau. Le deuxième rôle est de trouver le chemin le plus court et de s'y adapter [175].

Des hybridations entre Voronoi et l'optimisation des essaims de particules PSO sont conçues pour réaliser le même objectif qui consiste à maximiser la couverture et à prolonger la durée de vie d'un réseau. Les auteurs dans [176,177] ont utilisé le DV pour détecter les trous de couverture et pour évaluer la fonction de fitness, alors qu'ils ont utilisé le PSO pour déterminer la prochaine position des nœuds capteurs dans un champ d'intérêt donné. Ces études sont basées sur l'utilisation d'un nœud centralisé pour recueillir des informations sur les positionnements de tous les nœuds, sachant que chaque nœud a une connaissance préalable de son propre emplacement ainsi que de tous les autres nœuds. Ceci permet d'effectuer par la suite les calculs pour suggérer de nouveaux emplacements pour chaque nœud du réseau. Généralement la fonction fitness peut se concentrer soit sur la minimisation des trous de couverture, soit sur la minimisation de la consommation d'énergie.

Selon la littérature disponible actuellement, il existe peu d'études qui ont combiné un AG avec le DV pour résoudre le problème de déploiement et de réallocation de réseau d'objets de collecte de données. Ce choix a été utilisé avec un certain succès pour réaliser le même objectif afin d'améliorer la couverture du réseau et de prolonger sa durée de vie. Toutefois ces approches se basent sur l'utilisation de la fonction fitness développée par un AG pour évaluer les solutions candidates et sélectionner par la suite les prochaines positions des nœuds. Le DV est utilisé uniquement pour détecter les trous de couverture. Les solutions générées par AG sont conçues à l'intérieur des cellules de DV. Les nœuds constituant le réseau de collecte d'objets sont distanciés de leurs voisins pour minimiser la couverture des trous [177,171]. Parfois de nouveaux nœuds sont obligatoirement ajoutés aux emplacements spécifiés pour couvrir les trous [178]. Ces approches ont été réalisées en utilisant des connaissances sur les emplacements des nœuds.

## **2.7. Types d'architecture parallèles et distribuées pour les algorithmes d'optimisation**

### **2.7.1. Cas de l'algorithme génétique**

Selon la littérature récente [179,180], dans les algorithmes génétiques AG, des nombreux opérateurs peuvent être exécutés les uns indépendamment des autres. L'efficacité de l'AG, parallèle est d'atteindre la solution souhaitée dans les plus brefs délais avec une meilleure performance. Les AG parallèles sont fréquemment utilisés pour résoudre des problèmes de grande taille et promettent des gains de performances substantiels. La plupart de ces algorithmes ont été exécuté sur des machines parallèles massives et leur efficacité dépend du système de calcul parallèle. Dans beaucoup de ces problèmes, l'évaluation de la fonction de fitness pour chaque solution candidate, peut être calculée de manière indépendante. Ceci revient à dire

que chaque solution candidate peut être calculée en même temps, c'est-à-dire en parallèle avec les autres.

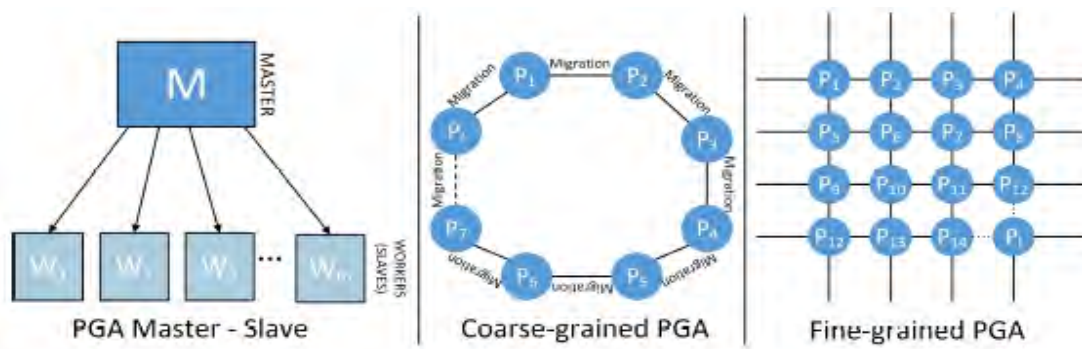
L'objectif principal de l'utilisation de cette architecture parallèle est de réduire le temps d'exécution et les ressources requises par l'AG. Des opérations génétiques sur des individus tels que le croisement, la mutation et l'évaluation de la population peuvent être réalisées en parallèle. L'idée principale de la plupart des programmes parallèles consiste à diviser les tâches en des sous-tâches plus réduites et à les exécuter simultanément sur différents nœuds. Cette approche peut être appliquée aux algorithmes génétiques de différentes manières.

La différence fondamentale entre ces différentes implémentations est d'utiliser une seule population ou de la subdiviser en des sous populations. Selon [181], les PGA (Parallel Genetic Algorithm) peuvent être classés en trois catégories principales comme la montre figure 2.11 :

- Dans le **modèle Master-Slave** [182,183], il existe un nœud maître pour gérer toutes les sous-populations et répartir les individus entre les nœuds esclaves. Par la suite, les valeurs de fitness des individus sont calculées dans les nœuds esclaves correspondants.

-Le **modèle Coarse-Grained**, est un modèle distribué [183], où la population est divisée en plusieurs sous-populations situées dans plusieurs "îles". Par la suite l'AG fonctionne indépendamment dans chaque "île". Ainsi, chaque île contient seulement des individus partiels de la population. Ces îles échangent périodiquement des informations en permettant à certains individus de immigrer afin d'injecter de la diversité dans la population [184,185]. Le modèle peut exécuter tous les opérateurs d'AG d'une manière parallèle et est même distribué dans les différentes « îles ».

-Le modèle **Fine-Grained**, nommé également le modèle cellulaire ou modèle de grille, il consiste à structurer une population en quartiers et à placer une partie d'individus dans un nœud. L'AG est réalisée en calcul parallèle pour évaluer la valeur de fitness de chaque chromosome et pour appliquer localement les opérateurs AG c'est-à-dire la sélection, le croisement et la mutation aux voisins adjacents. Sur la base d'une architecture massivement parallèle, le modèle peut considérablement accélérer l'évaluation de tous chromosomes. Cependant, cela nécessite un système de clustering massif pour gérer ce modèle [186].



**Figure 2. 11 Types d'Algorithme Génétique Parallèle [187]**

Notre objectif est d'élaborer une approche distribuée valable avec notre première hybridation entre le diagramme de Voronoi et l'AG et valable pour une deuxième hybridation entre la Triangulation de Delaunay et l'algorithme ABC qui seront exécutées parallèlement dans plusieurs nœuds. Notre idée est de répartir les calculs sur l'ensemble des nœuds du réseau, et de prendre en compte les aspects protocolaires, ce qui est rarement fait

### 2.7.2. Approche distribuée dans les RCD

Nous assistons depuis quelques années à l'émergence d'une nouvelle thématique sur l'utilisation des approches distribuées pour améliorer la couverture dans les réseaux sans fil de collecte de données.

Dans [188], les auteurs ont conçu un algorithme de barrière distribuée (DBarrier) basé sur le modèle de force virtuelle. Pour former une barrière, l'algorithme proposé laisse les nœuds capteurs se rassembler dans une certaine position à proximité d'une ligne traversant la limite gauche à la limite droite de la zone surveillée. Les nœuds capteurs ajustent leurs positions dans une barrière traversant la zone donnée en fonction des deux types de forces, c'est-à-dire les forces répulsives et attractives. La force virtuelle est calculée sur un seul nœud capteur, alors que les forces exercées seront exécutées d'une manière parallèle aux nœuds voisins. Le nœud capteur collecte les informations des voisins à deux sauts, à partir desquelles les voisins à un saut sont choisis en fonction de nouvelles positions après leur déplacement virtuel. Notons que cet algorithme ne peut former qu'une couverture à 1 barrière et ne fournit pas de solution efficace pour la couverture à k-barrières.

En [189], les auteurs se sont concentrés sur le problème de la relocalisation des nœuds capteurs pour construire une k-barrière avec une consommation d'énergie minimale. Les auteurs ont proposé un algorithme distribué nommé **AHVGB** (Approximate to Horizontal and Vertical Grid Barrier) pour construire une couverture de k-barrière dans des réseaux de capteurs à grande échelle. Premièrement, leur algorithme divise la zone du réseau en sous-réseaux de taille égale de sous-régions. Ensuite, les auteurs construisent une barrière de grille horizontale et une barrière de grille verticale dans chaque sous-région. Chaque sous-

région peut former indépendamment des barrières par des informations locales dans sa propre zone sans communication avec d'autres sous-régions. Bien que l'AHVGB puisse minimiser l'énergie de la communication et du mouvement, il ne peut pas maximiser le nombre d'obstacles.

Ahmed et al. [190] ont développé un algorithme de couverture probabiliste (PCA) pour évaluer la couverture dans un réseau de capteurs déployé aléatoirement. Les auteurs supposent que la propagation du signal d'une cible à un nœud capteur suit un modèle probabiliste. L'algorithme PCA est distribué et s'exécute sur tous les nœuds du réseau. Si un trou de couverture a été identifié, les nœuds mobiles peuvent être guidés vers des emplacements spécifiques pour satisfaire une détection contrainte de l'approche probabiliste. Cette approche est une déviation de l'hypothèse idéaliste du disque circulaire uniforme pour la couverture de détection utilisée dans le modèle de détection binaire. Les résultats de la simulation montrent que la couverture de zone calculée à l'aide de PCA est plus précise que la modèle de détection binaire idéaliste.

Selon la littérature, un grand nombre d'algorithmes est basé sur la détermination d'un ensemble dominant connecté CDS (Connected Dominating Set). Cet ensemble a un nombre minimum de nœuds nécessaires et suffisants pour garantir la couverture d'une zone d'intérêt.

Un algorithme distribué est proposé dans [191] nommé algorithme CDS-Rule-K. Il construit un CDS en deux phases et débute par la première phase où tous les nœuds diffusent un message "hello" pour recevoir des tables voisines et échangent leurs tables voisines. Si un nœud voisin n'est pas couvert par d'autres nœuds, il est marqué comme un nœud de CDS. La deuxième phase utilise des règles d'élagage pour couper les nœuds feuilles redondants. La règle d'élagage spécifie que si tous les nœuds adjacents sont couverts par des nœuds marqués, alors le nœud est un nœud feuille redondant, il est donc élagué et diffuse un message de mise à jour.

Selon les études que nous avons citées, le traitement des données dans les approches distribuées ne se fait pas dans des centres de calcul mais dans des nano-ordinateurs. Ces études introduisent la répartition de la charge de travail en utilisant plusieurs ressources de traitement. Le traitement de certaines parties de la charge de travail dans des différentes ressources pourrait résoudre ce problème.

Notre idée sur l'utilisation d'une approche distribuée pour le déploiement dans un réseau de collecte de donnée est de partager les tâches aux différents nœuds utilisés dans un réseau afin de gagner du temps, et augmenter globalement la durée de vie du réseau. L'exécution de ces tâches se fait en parallèle sur les différents nœuds.

## 2.8. Conclusion

En bilan, nous venons dans ce chapitre de décrire le problème de déploiement optimisé dans les réseaux sans fil de collecte des données. En premier lieu, nous avons présenté la couverture et la connectivité comme étant des facteurs très

importants dans la stratégie de déploiement. Nous avons présenté les techniques classiques de déploiement telles que la force virtuelle et les techniques basées sur la géométrie computationnelles comme le diagramme de Voronoi et la Triangulation de Delaunay. Dans un deuxième temps, nous avons élaboré une conception plus détaillée des méthodes de déploiement basées sur la méta-heuristique telles que l'algorithme génétique et l'algorithme de colonie des abeilles artificielles avec leur fonctionnement. En troisième lieu, nous avons présenté des hybridations entre les méthodes géométriques de déploiement et les algorithmes d'optimisation tel qu'il le cas et de comme l'Hybridation entre le Diagramme de Voronoi et l'algorithme génétique. Et enfin, nous avons donné une piste de recherche sur les approches distribuées et parallèles dans les algorithmes d'optimisations, nous y avons examiné le cas de l'algorithme génétique.

Dans le chapitre suivant, nous nous intéresserons à la présentation de notre démarche d'hybridations entre les méthodes géométriques de déploiement et les algorithmes d'optimisation par une approche centralisée.

# **Partie II: Contributions**



### **3. Proposition d'approches centralisées pour les hybridations: stratégies et résultats**





### 3.1. Introduction

Dans ce chapitre, nous proposons une approche centralisée maximisant la couverture d'une région d'intérêt par l'application des deux hybridations. La première hybridation nommée DVAG combine une méthode géométrique de déploiement présentée par le Diagramme de Voronoi DV, avec un algorithme d'optimisation présenté par l'algorithme génétique AG. Le DV a pour but de diviser le champ d'intérêt en cellules et d'engendrer des solutions initiales sur les positions de chaque nœud, puis un algorithme génétique AG est utilisé pour déterminer les meilleurs emplacements, à partir de résultats obtenus par le DV, afin de maximiser la couverture.

Une deuxième hybridation nommée TDABC combine une méthode géométrique de déploiement présentée par le diagramme de Triangulation de Delaunay TD, avec un algorithme d'optimisation présenté par l'algorithme de colonie des abeilles artificielles ABC. Le TD a pour but de générer des solutions qui présentent un déploiement initial des objets IoT, puis un algorithme ABC sera exécuté pour améliorer le taux de couverture de ce déploiement des nœuds.

Nous le verrons en fin de chapitre, les résultats révèlent que les hybridations centralisées de DVAG et TDABC dépassent en performances celles des algorithmes originaux en termes de degré de couverture, RSSI (Received Signal Strength Indicator), FER (Frame Error Rate) et nombre de voisins d'objets.

### 3.2. Contribution 1: première hybridation DVAG

Cette contribution DVAG- centralisée fait l'objet de notre publication [192].

#### 3.2.1. Algorithme proposé

Dans notre approche de déploiement réseau, nous supposons que tous les nœuds capteurs sont du même type dans une région d'intérêt. Comme première étape, notre algorithme DVAG commence par générer des solutions de Voronoï. Ces solutions présentent un premier déploiement des nœuds dans un champ d'intérêt, et en même temps, ces solutions présentent la population initiale de l'algorithme génétique AG. Chacune de ces solutions sera évaluée par le calcul de sa couverture. Nous avons utilisé par la suite l'AG pour améliorer notre résultat par le redéploiement des nœuds. Ce processus sera répété jusqu'à ce que la couverture cherchée soit trouvée (un taux minimal prédéfini de couverture est atteint). Un pseudo code schématise l'algorithme DVAG dans l'algorithme 3-1.

```

0) Définir la fonction d'évaluation f (Couverture maximale cherchée) ;
1) Pour i de 1 a nombre M des solutions

    Solution[i] ← Générer par le Diagramme de Voronoi
    Calcul de couverture (fonction fitness) de chaque solution Voronoi
    Fin Pour
2) Appliquer l'AG pour les M solutions Voronoi
    Répéter
    a) Sélection
    b) Croisement
    c) Mutation

    Jusqu'à une couverture maximale cherchée ou à un nombre d'itération fixé dès le début
3) Retourner la meilleure Solution trouvée
4) Affectation de position de chaque nœud
    
```

**Algorithme 3- 1 Pseudo-code de DVAG centralisée**

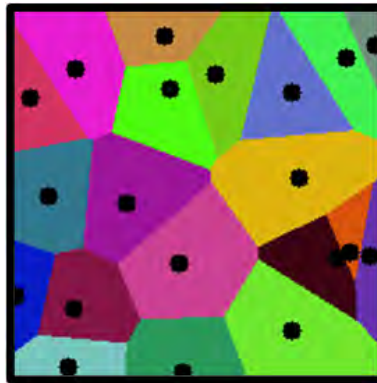
### **3.2.1.1. Diagramme de Voronoi**

#### **3.2.1.1.1. Processus de calcul**

Soit S un ensemble fini de n points du plan, une région de Voronoï ou bien cellule de Voronoï associée à un élément p de S, est l'ensemble des points qui sont plus proches de p que de tout autre point de S comme le montre la figure 3.1.

$$Vors(p) = \{x \in \mathbb{R}^2 \ \forall q \in S \ \|x - p\| \leq \|x - q\|\} \quad (11)$$

Avec  $\|x - p\|$  désigne la distance entre x et p.

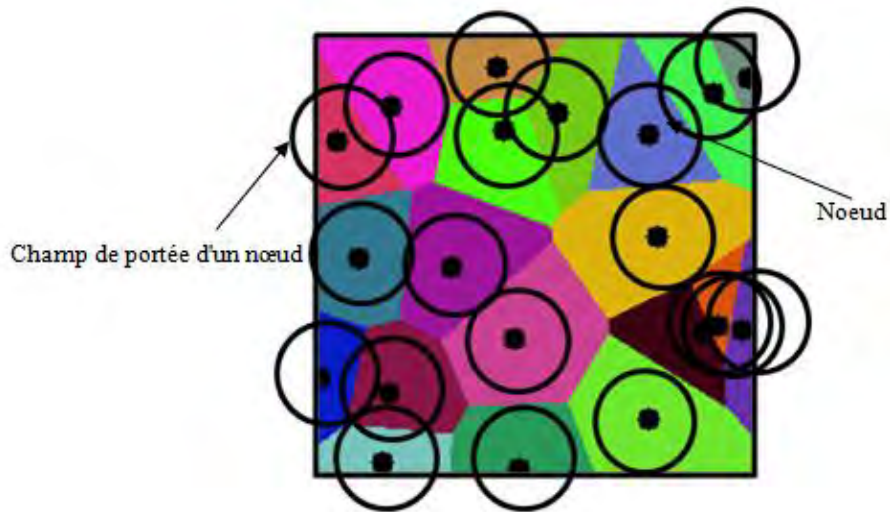


**Figure 3. 1 Diagramme de Voronoi**

#### **3.2.1.1.2. Déploiement initial avec DV**

Dans la plupart des travaux cités dans le chapitre 2, le diagramme de Voronoi est utilisé pour détecter les trous de couverture dans un champ d'intérêt après un déploiement initial des nœuds. Dans notre approche nous avons utilisé le DV pour subdiviser le champ en des cellules, ensuite chaque nœud capteur sera affecté à une seule cellule de ces cellules de Voronoi. Le point qui définit chaque cellule présente

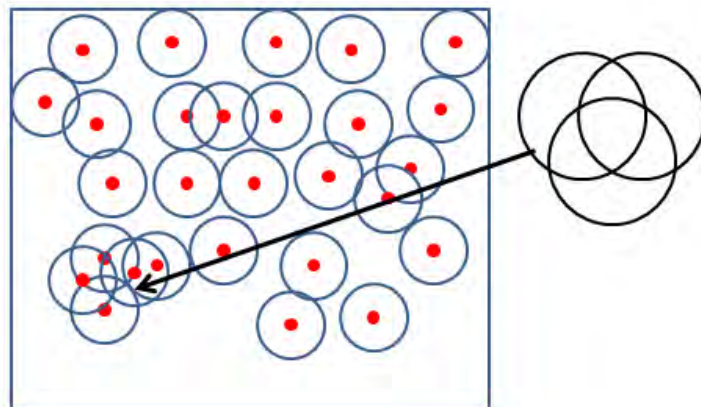
les coordonnées d'un nœud capteur. Comme la montre figure 3.2, chaque nœud capteur se trouve dans une cellule de Voronoi.



**Figure 3. 2 Exemple d'une solution d'un déploiement initial des objets par le DV**

Par la suite, notre algorithme permet d'engendrer un nombre  $M$  de solutions de Voronoi, chaque solution contient les positions de  $N$  nœuds déployé sur une région donnée. Le nombre des solutions présente le nombre d'itération de DV que l'on peut le fixer dès le début.

Dans le but de garantir une couverture maximale de la région d'intérêt avec le moins de chevauchement de couverture entre les différents nœuds, nous avons éliminé les solutions de Voronoi qui contiennent plus de deux chevauchements de nœuds comme le montre la figure 3.3 et ceci dans le but de simplifier prochainement la tâche d'amélioration de ces positionnements par l'algorithme d'optimisation AG.



**Figure 3. 3 Suppression des solutions de Voronoi qui ont plus de deux nœuds en chevauchement [192]**

### 3.2.1.2. Algorithme génétique

Dans cette partie nous appliquons l'AG standard avec quelques modifications appropriées expliquées dans les sous-sections suivantes. Le processus de notre AG commence par un ensemble d'individus (ou solutions) composées d'un ensemble de solutions possibles générées par le diagramme de Voronoï dans notre approche. Ces solutions de Voronoï représentent la population de notre AG. Ce processus génétique est répété jusqu'à ce qu'une condition d'arrêt soit satisfaite (la couverture minimale désirée).

#### 3.2.1.2.1. Codage des chromosomes

Dans un algorithme génétique, un individu est défini par un chromosome qui représente généralement une solution. Dans le codage des chromosomes, nous avons utilisé un tableau à 2 dimensions pour présenter les coordonnées de chaque nœud. Chaque coordonné représente les positions  $(x,y)$  d'un nœud donné par le diagramme de Voronoï comme le montre figure 3.4

	Noeud1	Noeud 2				Noeud n	
$x$	128.32	78.96	186.03	20.56	...	...	105.20
$y$	17.31	149.69	89.14	66.13	...	...	59.78

Figure 3. 4 Représentation d'un chromosome dans un champ 200\*200

#### 3.2.1.2.2. Population Initiale

Le processus de l'algorithme génétique commence par la population initiale qui représente l'ensemble de tous les individus qui sont utilisés pour trouver la meilleure solution. Dans notre approche le résultat donné par le Diagramme de Voronoï est la population initiale de l'algorithme génétique qui représente M solutions de Voronoï c'est-à-dire M individus en entrés pour l'AG. Par la suite la taille de la population est M individu (M solutions) comme le montre l'exemple dans la figure 3.5.

$x_1$	$y_1$	$x_2$	$y_2$	.....	$x_M$	$y_M$
128.32	17.31	10.23	88.28		122.60	55.10
78.96	149.69	155.18	100.02		68.96	82.41
186.03	89.14	63.28	195.30		128.12	80.99
20.56	66.13	17.36	128.66		53.26	17.85
...	...	...	...		...	...
105.20	59.78	139.20	69.80		182.86	13.20
<b>Individu 1</b>		<b>Individu 2</b>			<b>Individu M</b>	

Figure 3. 5 Représentation de la population initiale

### 3.2.1.2.3. Fonction fitness

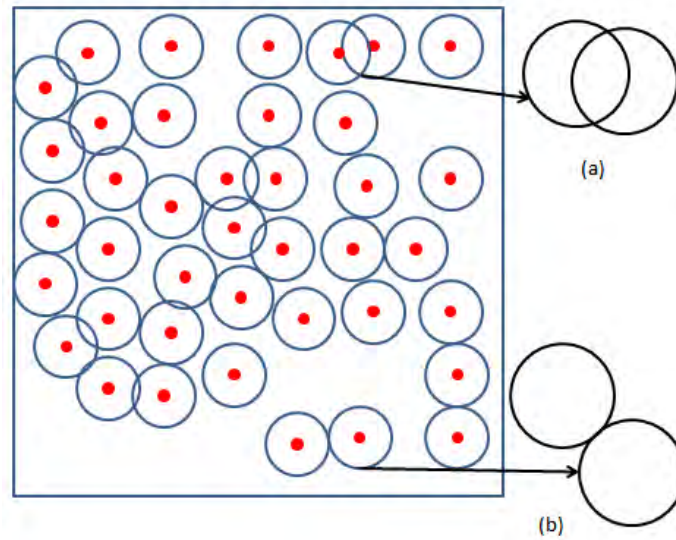
L'objectif de notre approche est de déterminer les meilleures solutions qui garantissent une couverture maximale dans une région d'intérêt. La fonction fitness, peut être présentée mathématiquement dans l'équation (12), est utilisée généralement dans l'AG pour déterminer les meilleures solutions.

$$Fitness = S - \sum_{i=0}^n \text{surfaces de tous les cercles} \quad (12)$$

Où  $S$  la surface de champs d'intérêt et  $n$  le nombre des nœuds à déployer; plus la différence entre la surface  $S$  et la somme des surfaces des nœuds diminue, plus la couverture augmente.

Dans le cas où il n'y a pas de chevauchement entre deux nœuds leur distance est égale ou supérieure à  $2r$ , comme illustré dans la figure 3.6 (b)

$$X = 2r - \text{distance} (p1 - p2) \quad (13)$$



**Figure 3. 6 Chevauchement des nœuds [192]**

Dans le cas où il y a un chevauchement entre deux nœuds comme illustré dans la figure 3.6 (a), leur distance l'un de l'autre est inférieure à  $2r$ . L'aire de la lentille  $L$  qui présente l'espace d'intersection entre 2 nœuds est égale à deux fois celle du segment  $MO'M'$  comme le montre la figure 3.7.

Aire du segment:  $A_S = 2R^2 \arccos\left(\frac{h}{R}\right) - 2h \sqrt{R^2 - h^2}$

$$A_L = 2R^2 \arccos\left(\frac{h}{R}\right) - 2h \sqrt{R^2 - h^2} \quad (14)$$

$$A_L = 2R^2 \arccos\left(\frac{d}{2R}\right) - d \sqrt{R^2 - \left(\frac{d}{2}\right)^2} \quad (15)$$

Où  $d$  est la distance entre les deux centres et  $h = d/2$ .

Le cas où il y a plus que deux nœuds qui se chevauchent est déjà éliminé par le diagramme de Voronoï.

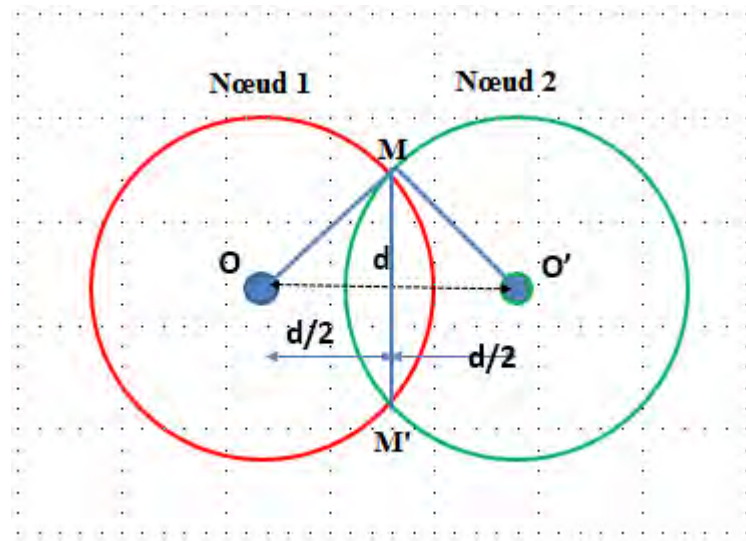


Figure 3. 7 Calcul d'aire des nœuds qui se chevauchent

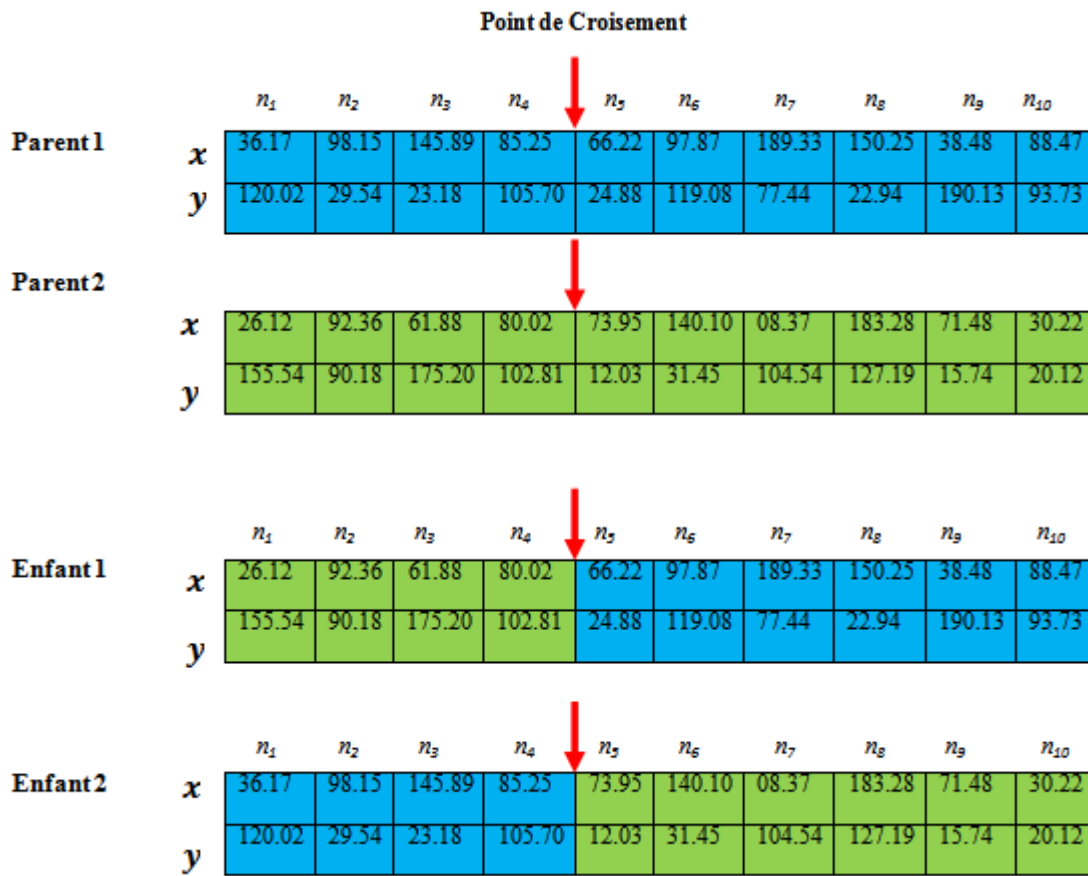
#### 3.2.1.2.4. Sélection

Après la première phase de la population initiale et pour reproduire de nouvelles générations nous avons appliqué le mécanisme de sélection de tournoi [206, 207] dans le but de sélectionner les individus. L'objectif de cette phase est de sélectionner les individus les plus aptes pour passer leurs gènes à la prochaine génération. Dans notre algorithme DVAG les individus les plus aptes sont les solutions qui ont un taux de couverture plus élevé que les autres solutions.

#### 3.2.1.2.5. Croisement

A partir de deux parents sélectionnés de la population, nous avons appliqué un croisement à un point au hasard. Cette opération de croisement est illustrée sur la figure 3.8 où le deuxième parent échange son premier segment de gène avec le premier segment de parent 1, et ceci dans le but de produire un nouvel individu (nouvelle solution). La figure 3.8 présente un exemple de déploiement de 10 nœuds de  $n_1$  à  $n_{10}$  dans un champ  $200 \times 200$ .





**Figure 3. 8 Croisement par un point**

Tous les individus dans la population initiale peuvent être, avec une probabilité de croisement, des parents possibles pour les futures « Enfants », ainsi le processus de la roulette est utilisé pour sélectionner le deuxième parent de la population pour appliquer la phase de croisement.

### **3.2.1.2.6. Mutation**

Une nouvelle population est formée par N individus après l'utilisation des opérateurs de sélection et de croisement. Chaque individu va participer à une mutation en un point. Généralement l'emplacement du gène mutant qui va être remplacé par une valeur différente est choisi au hasard. Un exemple sur l'opérateur de mutation est illustré dans la figure 3.9, où l'emplacement du gène mutant est égal à 6 donc, la position de nœud 6 va être changée.

	$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_7$	$n_8$	$n_9$	$n_{10}$
<b>Parent</b>	26.12	92.36	61.88	80.02	66.22	97.87	189.33	150.25	38.48	88.47
	155.54	90.18	175.20	102.81	24.88	119.08	77.44	22.94	190.13	93.73
<b>Enfant</b>	26.12	92.36	61.88	80.02	66.22	31.92	189.33	150.25	38.48	88.47
	155.54	90.18	175.20	102.81	24.88	132.16	77.44	22.94	190.13	93.73

Figure 3. 9 Opérateur de mutation en un point

### 3.2.1.2.7. Double mutation

La figure 3.10 montre le même exemple que la figure 3.9 avec double mutation, ce qui veut indiquer que, chaque individu va participer à une mutation en deux points. Deux objets seront repositionné le nœud 3 (66.88, 175.20) sera à la position (27.50, 102.60) et au même temps le nœud 8 (150.25, 22.94) sera à la position (67.90, 159.10).

	$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_7$	$n_8$	$n_9$	$n_{10}$
<b>Parent</b>	26.12	92.36	61.88	80.02	66.22	97.87	189.33	150.25	38.48	88.47
	155.54	90.18	175.20	102.81	24.88	119.08	77.44	22.94	190.13	93.73
<b>Enfant</b>	26.12	92.36	27.50	80.02	66.22	97.87	189.33	67.90	38.48	88.47
	155.54	90.18	102.60	102.81	24.88	119.08	77.44	159.10	190.13	93.73

Figure 3. 10 Opérateur de mutation en deux points

Pour le critère de terminaison de l'AG, nous pouvons utiliser une condition d'arrêt sur le nombre d'itérations ou une condition d'arrêt sur le taux de couverture cherché.

### 3.2.2. Simulation

Pour évaluer notre algorithme DVAG dans cette approche centralisée, nous avons utilisé le logiciel de simulation de réseaux OMNET++ [196]. Tout d'abord, nous avons appliqué le DV sur un ROI de 100\*100 mètres, le DV divise la région en un ensemble de cellules de Voronoi pour générer une population de 100 solutions de Voronoi (individus) qui représentent chacune une position d'un objet IoT dans un plan 2D. Les solutions données par le DV représentent la population initiale de l'AG.

L'AG permet d'optimiser les solutions trouvées par le DV en remplaçant les positions des nœuds afin de leur trouver le meilleur positionnement dans le RoI et d'avoir par la suite un degré de couverture maximal. Le choix des valeurs des paramètres repose sur des tests empiriques et sur nos études précédentes détaillées dans [192]. Le tableau 3.1 montre les paramètres de simulation utilisés. L'utilisation des paramètres taux de croisement et taux de mutation sont justifiés dans [204]

**Tableau 3. 1 Paramètres de simulation de VDAG**

<b>Paramètres</b>	
Taille de la population: Nombre des individus	100
Nombre d'itération	[100..1500]
Nombre des objets à déployer (nœuds)	40
Taux de croisement	0.9
Taux de mutation	0.1
Taille ROI	100*100
Condition d'arrêt	94%
RSSI	100 (initial)
FER	0.01 (initial)
Portée de transmission intérieure	10 m
Portée de détection intérieure	10 m
Fréquence	2.4 GHz
Moyenne du nombre d'exécution	30
Antenne	isotropique
Température de fonctionnement du nœud	0° C à 60° C

### 3.2.3. Résultats et discussions

Nous présentons dans cette partie les résultats obtenus de l'hybridation DVAG dans une approche centralisée avec des comparaisons en termes de taux de couverture, RSSI, durée de vie et nombre de voisins.

**DV vs DVAG :** La figure 3.11 et la figure 3.12 montrent la différence, en termes de degré de couverture, entre la couverture aléatoire initiale générée uniquement par le diagramme de Voronoi et le degré de couverture fourni par le DVAG proposé. En effet, nous remarquons que la répartition des nœuds obtenu par l'algorithme DVAG (figure 3.12) est meilleure (plus uniforme et avec des nœuds moins chevauchés) que la répartition des nœuds obtenu par l'algorithme DV (figure 3.11).

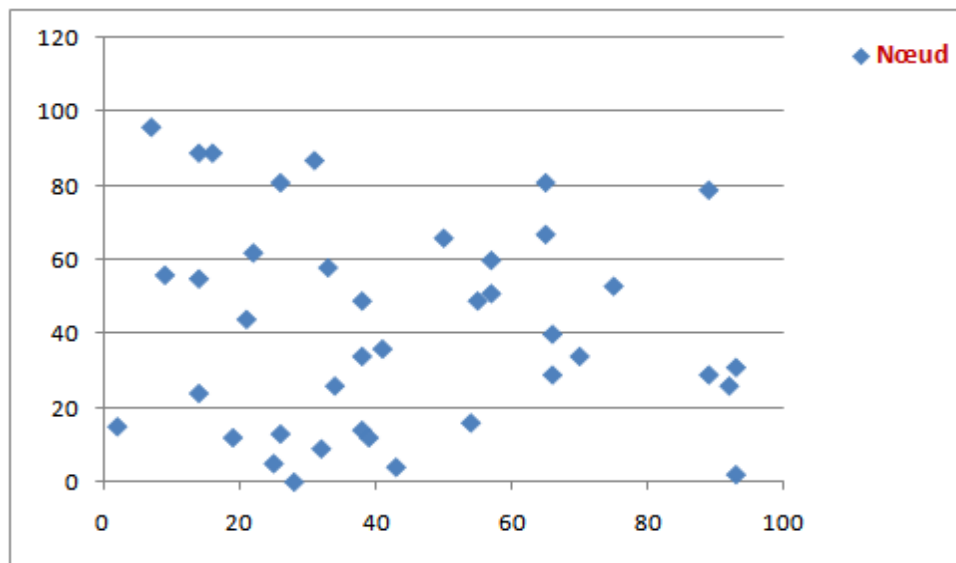


Figure 3. 11 Déploiement aléatoire initial en utilisant DV

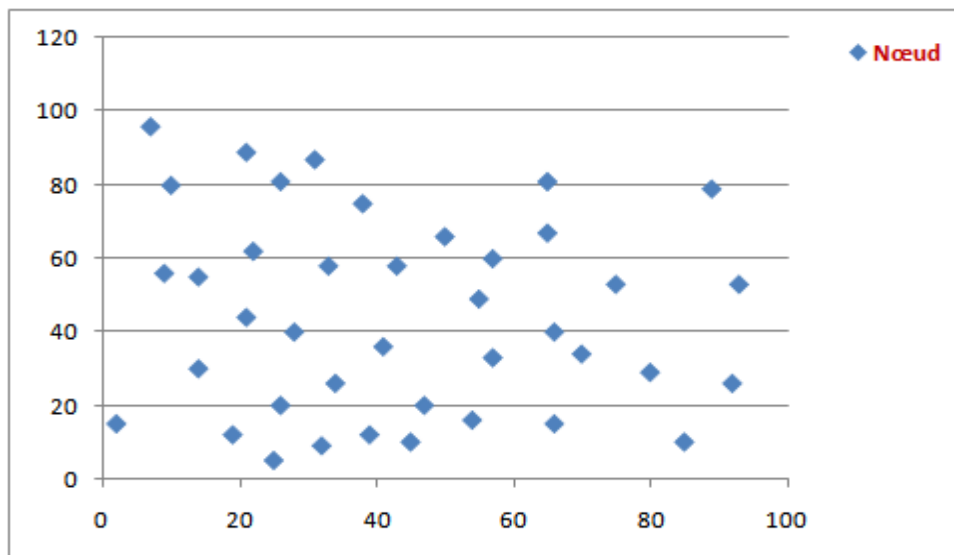
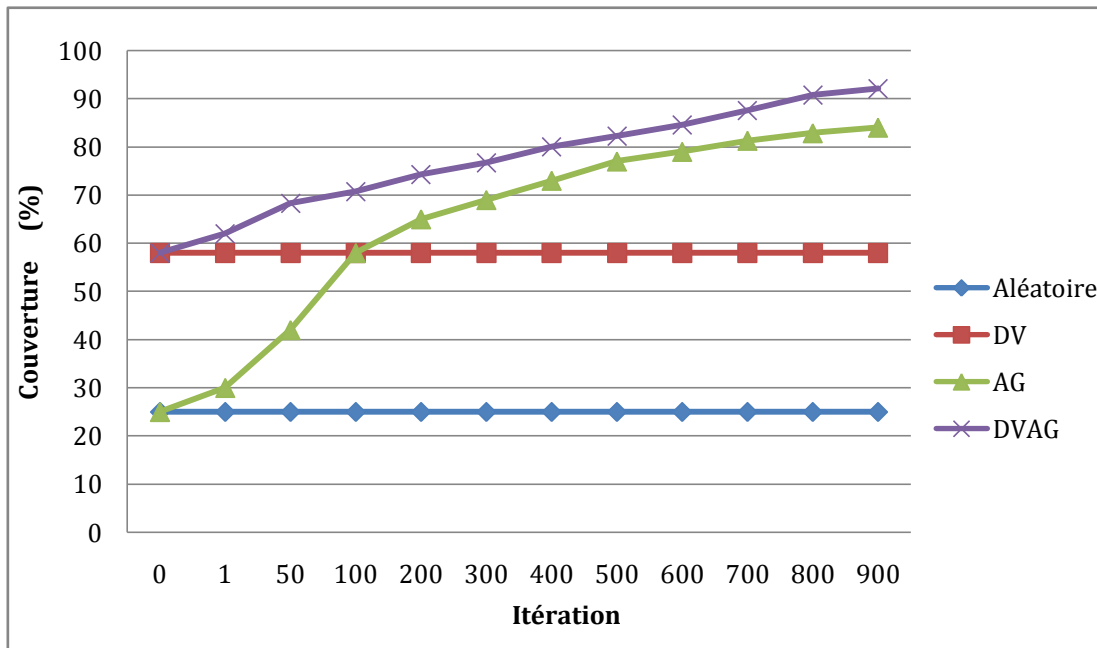


Figure 3. 12 Déploiement des nœuds après l'exécution de DVAG

**Taux de couverture** : La figure 3.13 permet de comparer le taux de couverture des algorithmes pour 25 exécutions en moyenne.

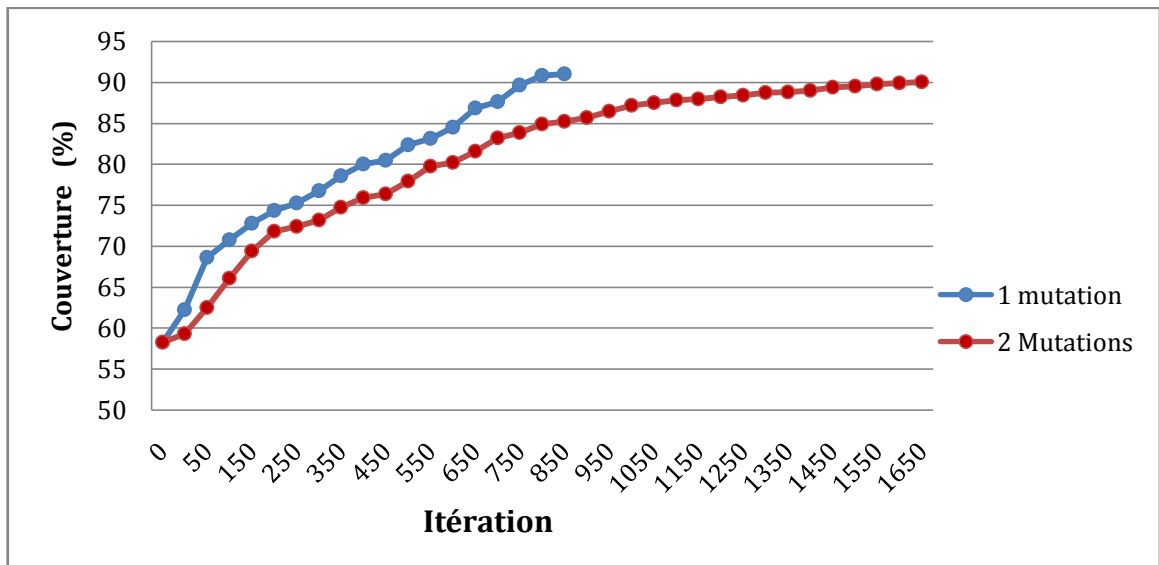


**Figure 3. 13 Comparaison du degré de couverture en fonction du nombre d'itérations pour DV, AG et DVAG**

La figure 3.13 montre que le degré de couverture est amélioré lorsque le nombre d'itérations pour AG augmente. De plus, la figure 3.13 démontre que le DV est meilleure que l'AG pour un nombre d'itérations de l'algorithme AG inférieur à 100, tandis que l'algorithme AG donne un meilleur degré de couverture pour plus de 100 itérations. L'algorithme hybride DVAG est meilleur que le DV et l'AG dès la première itération.

**Double mutation** : nous avons comparé également le résultat de notre hybridation DVAG dans une approche centralisée qui utilise une seule mutation avec DVAG en double mutations. A la place de changer la position d'un seul nœud, on change les positions des deux nœuds au même temps. La figure 3.14 montre qu'après l'exécution de notre hybridation DVGA dans une approche centralisée, un AG est utilisé avec une seule mutation et se termine après environ à 800 itérations, tandis qu'un AG avec deux mutation prend beaucoup plus de temps (il nécessite plus que 750 itérations supplémentaires pour atteindre une couverture de la surface de 90%).

L'utilisation de double mutations pour un individu augmente le caractère aléatoire, et donc augmente les solutions de mauvaise qualité (mauvaise couverture). C'est pour cela un AG avec deux mutations prend beaucoup plus de temps qu'un AG avec une seule mutation



**Figure 3. 14 Comparaison entre l'opérateur de mutation à un point et à deux points [192]**

**Nombre d'objectifs** : le tableau 3.2 présente les valeurs d'Hyper-volume (meilleur, moyen et pire) de 25 exécutions de l'AG, pour différents nombres d'objectifs et différents nombres d'itérations. L'Hyper-volume est une métrique d'optimisation qui permet d'évaluer la qualité des solutions fournis par l'algorithme en termes de dominance [197]. L'indicateur d'hypermolume est l'un des indicateurs de qualité d'ensemble les plus utilisés pour l'évaluation des problèmes d'optimisations multiobjectifs stochastiques, ainsi que pour la sélection dans les algorithmes d'optimisation multiobjectif évolutionnaire.

Ses propriétés théoriques justifient sa large acceptation, en particulier la stricte monotonie par rapport à la dominance de l'ensemble qui est encore unique des indicateurs basés sur l'hypermolume. L'évolution de l'hypermolume donne une bonne information sur l'évolution de la convergence de la population. Plus l'hypermolume augmente plus la solution est bonne

**Tableau 3. 2 Meilleure, moyenne et pire valeurs d'Hyper-volume de l'algorithme génétique**

Nombre d'objectifs	Nombre max d'itérations	AG
1	100	0.9393
		0.9264
		0.9031
	400	0.9495
		0.9422
		0.9247

2	100	0.9383 0.9296 0.8835
	400	0.9438 0.9399 0.8934

Le tableau 3.2 montre qu'en considérant un seul objectif (le degré de couverture) au lieu de deux objectifs (degré de couverture et coût de déploiement), l'AG donne des valeurs plus élevées (donc meilleures) d'Hyper-volume. De plus, augmenter le nombre d'itérations, permet à l'AG d'obtenir de meilleurs résultats (valeurs plus élevées de l'Hyper-volume).

Le tableau 3.3 compare le comportement des algorithmes testés concernant les valeurs de RSSI (indiquant la puissance du signal radio reçu) lors de la réception de trames échangées entre les nœuds du réseau, la durée de vie des nœuds, et le nombre moyen de voisins pour tous les nœuds. En effet, le RSSI donne une idée sur la qualité du signal transmis, puis il indique la qualité de la couverture. La durée de vie est définie par le temps (en secondes) au bout duquel le premier nœud s'éteint. Concernant le nombre de voisins de chaque nœud, un nœud  $i$  est considéré comme voisin d'un autre nœud  $j$  si le RSSI moyen entre  $i$  et  $j$  est supérieur à un seuil (fixé à 100 dans nos tests, on utilise des valeurs RSSI numériques de 8 bits (0-255) convertible en valeurs Dbm) et au FER maximum entre les deux nœuds ne dépasse pas un autre seuil (fixé à 0,1 dans nos tests, ce qui veut dire 1 trame reçue fautive toutes les 10 trames reçues).

**Tableau 3. 3 Comparaison des valeurs moyennes du RSSI, de la durée de vie et du nombre de voisins**

	DV	AG	DVAG
RSSI	132.39	131.83	137.28
Durée de vie(en secondes)	1298	1367	1693
Nombre des voisins	4.23	4.65	5.98

Le tableau 3.3 indique que l'algorithme hybride proposé est meilleur par rapport à l'utilisation d'AG seul et aussi meilleur par rapport à l'utilisation de DV seul. Une observation intéressante est que malgré les meilleures performances d'AG par rapport au DV concernant la durée de vie et le nombre de voisins, ce dernier algorithme donne de meilleurs résultats que l'AG concernant les valeurs de RSSI. Il faut noter que dans nos simulations, le RSSI est modéliser en particulier par l'affaiblissement

du signal le long du chemin entre les deux nœuds, ceci correspond donc principalement à la distance entre les nœuds.

Notre approche proposée montre une meilleure couverture par rapport à d'autres résultats [175, 178]. Dans [178], une optimisation d'algorithme génétique a été présentée avec trois scénarios différents avec une couverture de 90 %. Cependant, cette étude a utilisé un nombre élevé de nœuds mobiles et fixes pour parvenir à ces résultats. Notre approche proposée utilise moins de nœuds (donc une densité plus faible et donc plus réaliste) avec néanmoins meilleure couverture.

Afin de confirmer l'efficacité de notre algorithme, nous allons comparer la meilleure solution obtenue avec l'estimation du rapport d'approximation adaptative de DVGA, pour différents intervalles de temps, avec une complexité  $O(gnM)$ , où  $g$  est le nombre de générations,  $n$  est la taille de la population et  $M$  présente le nombre de nœuds mobiles.

### 3.3. Contribution 2 : seconde hybridation TDABC

#### 3.3.1. Algorithme proposé

Comme déjà dit précédemment, dans notre approche de déploiement réseau, nous supposons que tous les nœuds capteurs sont du même type dans une région d'intérêt. Comme première étape, notre algorithme DTABC commence par générer des solutions de Delaunay; ces solutions représentent un premier déploiement des nœuds dans un champ d'intérêt, et en même temps, ces solutions présentent la population initiale de l'algorithme de colonie des abeilles artificielles ABC. Chacune de ces solutions sera évaluée par le calcul de sa couverture. L'algorithme de colonie des abeilles artificielle a pour objectif d'améliorer notre résultat par le redéploiement des nœuds. Ce processus sera répété jusqu'à ce que la couverture cherchée soit trouvée. Un pseudo code schématise l'algorithme TDABC dans l'algorithme 3-2.

- 1) Définir la fonction d'évaluation  $f$  (Couverture maximale cherchée) ;
- 2) Générer  $N$  solution de Delaunay  $S_i = S_1, S_2, S_3, \dots, S_N$
- 3) Calculer la couverture de chaque solution  $S_i$
- 4) Evaluer la fonction fitness  $f_i$  de chaque solution
- 5) Cycle = 1;
- 6) Répéter
- 7)    Pour chaque abeille employée
- 8)    {
- 9)    Produire une nouvelle solution
- 10) Calculer la valeur  $f_i$
- 11) Appliquer le processus de sélection de glouton
- 12)    }
- 13) Calcul de probabilité  $P_i$  de chaque solution  $S_i$
- 14) Pour chaque abeille spectatrice
- 15)    {
- 16)    Sélection d'une solution  $S_i$  selon la probabilité  $P_i$
- 17)    Produire une nouvelle solution
- 18)    Calculer la valeur  $f_i$
- 19) Appliquer le processus de sélection de glouton
- 20)    }



- 21) S'il y a une solution abandonnée
- 22) Alors remplacez-la par une nouvelle solution produite aléatoirement
- 23) Mémoriser la meilleure solution
- 24) Cycle ++ ;
- 25) Jusqu'à cycle = Nombre de cycle défini ou jusqu'à une couverture maximale trouvée

### **Algorithme 3- 2 Pseudo code de TDABC centralisée**

#### **3.3.1.1. Triangulation de Delaunay**

Dans la plupart des travaux cités dans le chapitre 2, le diagramme de Triangulation de Delaunay TD est utilisé pour détecter les trous dans la couverture après un déploiement initial des nœuds. Dans notre approche nous avons utilisé le TD pour subdiviser le champ en des triangles. Les positions des nœuds correspondent aux coordonnées des centres des cercles circonscrits des triangles de Delaunay. Dans cette partie, nous commençons par détailler les étapes de calcul de TD selon l'algorithme de Watson [193] puis nous nous intéressons à l'algorithme ABC et son fonctionnement pour améliorer le taux de couverture dans un champ d'intérêt.

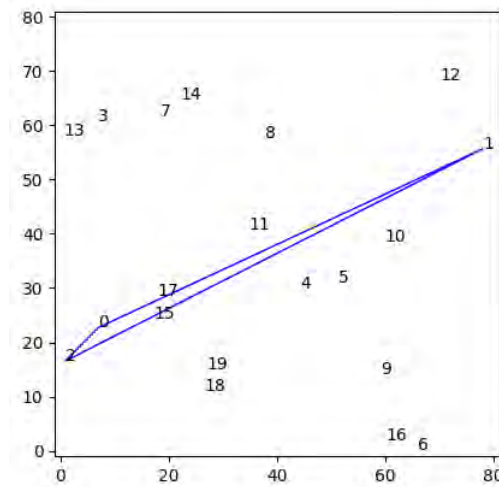
##### **3.3.1.1.1. Processus de calcul**

Nous avons suivi l'algorithme de Watson pour calculer la Triangulation de Delaunay (TD) d'un ensemble fini de points. Cette triangulation est assemblée en introduisant chaque point un par un. Le processus commence par trois points qui sont choisis pour former un "super-triangle" comme le montre la figure 3.15.

Lorsqu'un nouveau point est introduit dans une triangulation existante, le centre circonscrit de chaque triangle existant est calculé avec son rayon de cercle circonscrit. Tous les triangles qui contiennent le nouveau point dans leurs cercles circonscrits sont marqués comme étant intersectés. Chaque nouveau point doit intersecter au moins un des cercles circonscrits du triangle existant. Les triangles qui sont intersectés lorsqu'ils sont additionnés, forment un polygone. Tous les triangles intersectés sont remplacés, et le nouveau point forme des triangles avec chaque paire de sommets sur la limite du polygone (le nouveau point est toujours à l'intérieur du polygone)

Les étapes fondamentales de l'algorithme de Watson sont les suivantes :

0. Générer N points (qui présentent le nombre des nœuds à déployer) à trianguler aléatoirement dans un champ d'intérêt (l'exemple est dans un champ d'intérêt 80\*80)
  
1. Définir les sommets du super-triangle comme le montre la figure 3.15. Les sommets sont numérotés de 1 à N. Puis définir les coordonnées de ces sommets. Ajoutez le super-triangle à une liste de triangles formés et le marquer comme incomplet.



**Figure 3. 15 Générer N points aléatoire et définir les sommets du super triangle**

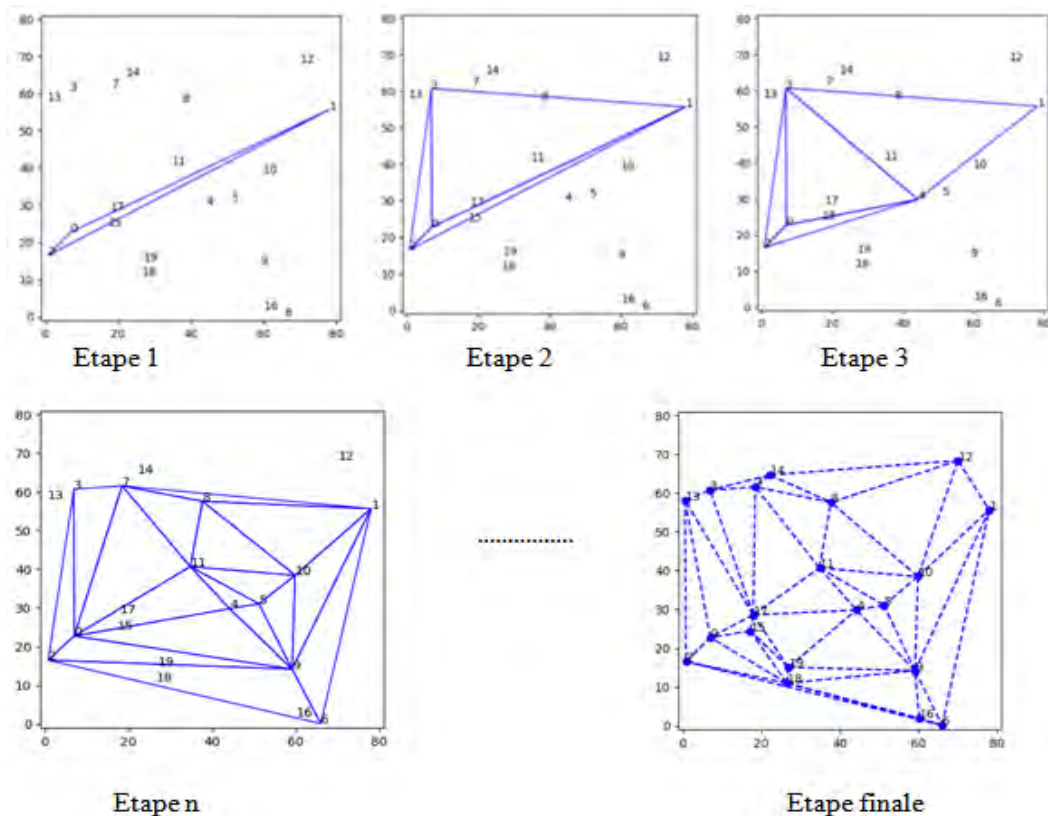
2. Introduire un nouveau point de la liste des points avec les coordonnées  $(x_{nv}, y_{nv})$
3. Examiner la liste de tous les triangles formés jusqu'à présent et pour chaque triangle signalé comme incomplet, effectuer les étapes 5 à 9
4. Calculer les coordonnées du centre de cercle circonscrit du triangle  $(x_c, y_c)$ , et le carré de son rayon du cercle circonscrit  $R^2$
5. Calculer le carré de la distance  $x$  entre le nouveau point et le centre circonscrit du triangle, c'est-à-dire la valeur  $d_x^2 = (x_c - x_{nv})^2$
6. Si  $d_x^2 > R^2$ , alors le cercle circonscrit de ce triangle ne peut être intersecté par aucun des points restants. Donc : marquer ce triangle comme terminé et n'exécuter pas les étapes 7 et 8
7. Calculer le carré de la distance entre le nouveau point et le centre circonscrit du triangle, c'est-à-dire  $d^2 = d_x^2 + (y_c - y_{nv})^2$
8. Si  $d^2 < R^2$ , alors le nouveau point se croise avec le cercle circonscrit pour ce triangle. Donc : supprimer ce triangle de la liste des triangles formés et stocker les trois paires de sommets qui définissent ses arêtes sur une liste d'arêtes.  
- Si  $d^2 > R^2$ , alors le nouveau point se trouve sur ou en dehors du cercle circonscrit pour que ce triangle reste inchangé.
9. Parcourir la liste des arêtes et supprimer toutes les arêtes qui apparaissent deux fois dans la liste. Cela supprime toutes les arêtes qui sont à l'intérieur du polygone formé par les triangles intersectés.

**10.** Former les nouveaux triangles en faisant correspondre le nouveau point avec chaque paire de sommets et le dernier des bords. Le nouveau point forme de nouveaux triangles avec chaque paire de sommets sur la limite du polygone formé par les triangles intersectés. Définir chaque nouveau triangle tel que ses sommets sont toujours répertoriés dans le sens inverse des aiguilles d'une montre. Ces nouveaux triangles sont signalés comme incomplets.

**11.** Répétez les étapes 2 à 10 jusqu'à ce que la liste des points, soit épuisée.

**12.** Former la triangulation finale en supprimant tous les triangles qui contiennent un ou plusieurs sommets du super-triangle. Ceci peut être réalisé en parcourant la liste des triangles et en supprimant tous ceux qui ont un nombre de sommets supérieurs à N.

La figure 3.16 schématise les étapes nécessaires pour calculer la Triangulation de Delaunay.



**Figure 3. 16** Processus de calcul de Triangulation de Delaunay

### 3.3.1.1.2. Déploiement initial avec TD

La figure 3.17 montre des différentes triangulations dans un même champ d'intérêt et avec le même nombre de points (N=20).

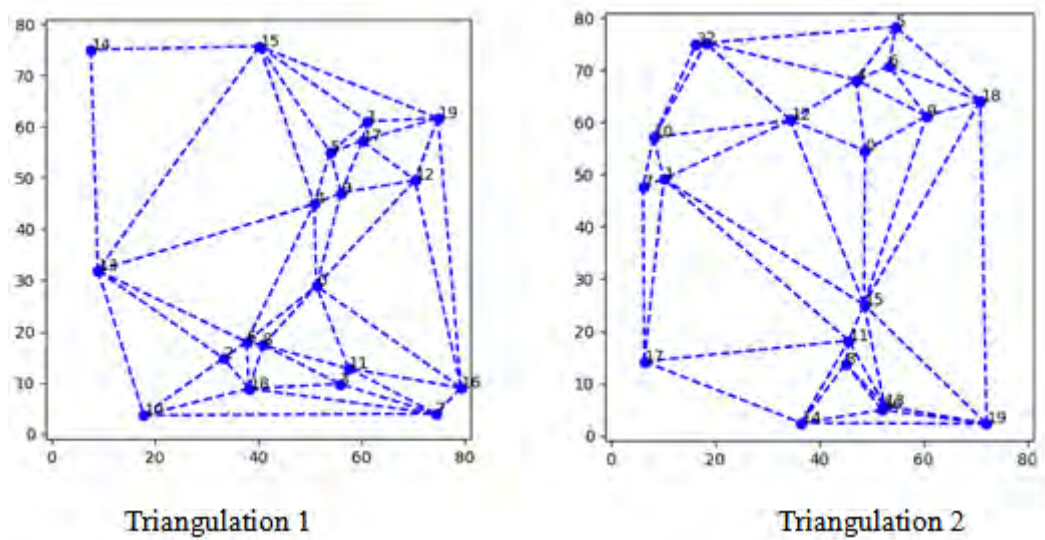


Figure 3. 17 Exemple de triangulation dans un champ 80\*80

Comme déjà noté dans le chapitre 2 à la section consacrée à la triangulation de Delaunay, la condition de Delaunay affirme qu'un réseau de triangles est une triangulation de Delaunay si tous les cercles circonscrits des triangles du réseau sont vides. La figure 3.18 montre une triangulation avec visualisation des cercles circonscrits aux des triangles : la condition de Delaunay est bien vérifiée.

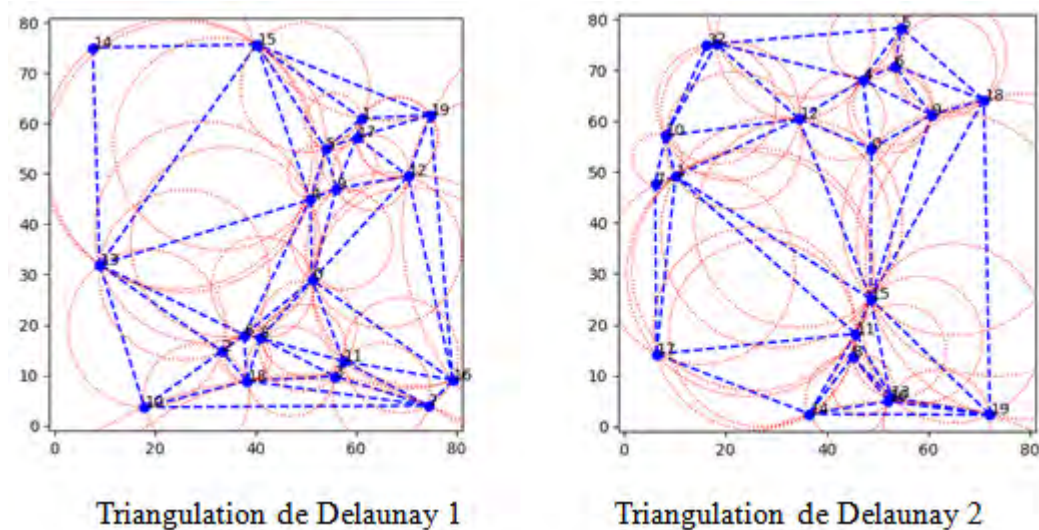
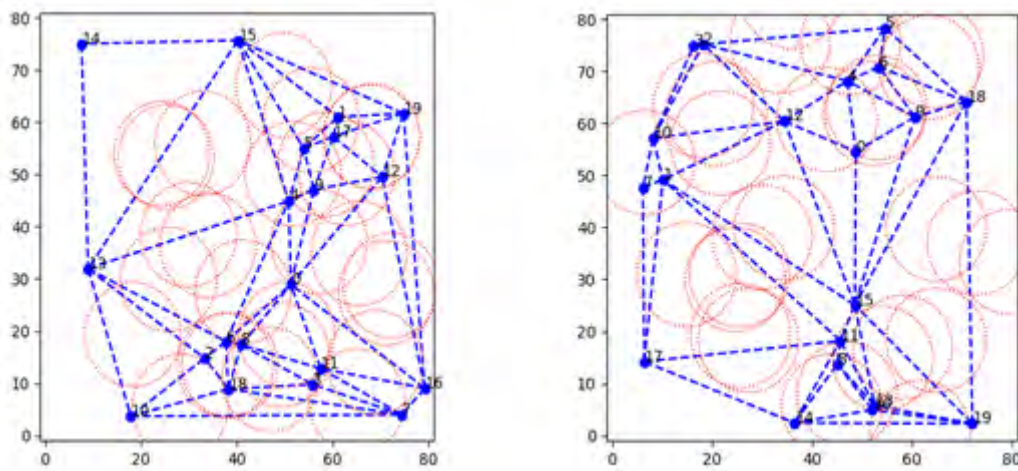


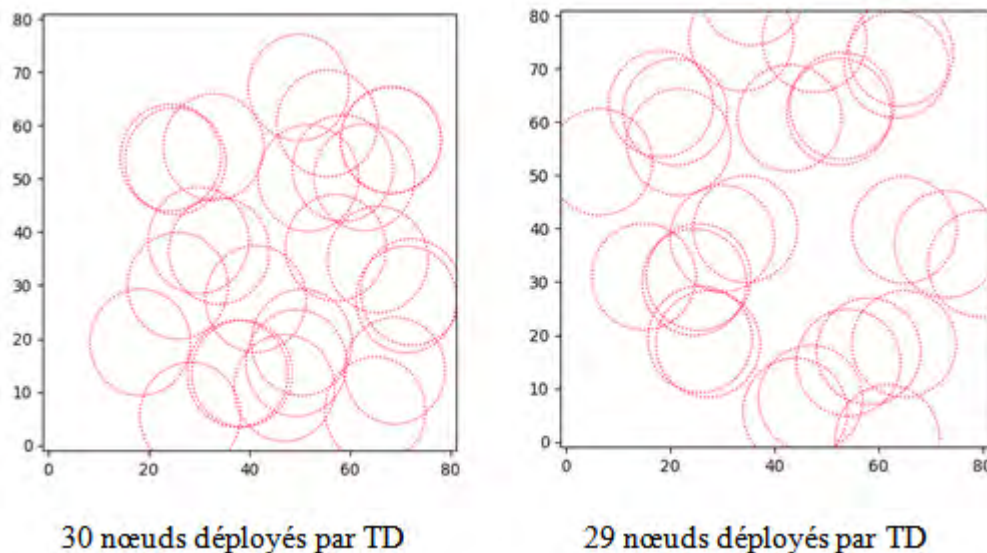
Figure 3. 18 Exemple de Triangulation de Delaunay dans un champ 80\*80

Dans notre propre déploiement initial des nœuds, nous avons considéré que la position de chaque nœud est la position du centre d'un cercle circonscrit de chaque triangle de la Triangulation de Delaunay. Et puisque nous avons supposé que tous les nœuds capteurs sont du même type dans une région d'intérêt (ils ont la même portée de détection et de communication) notre déploiement sera illustré sur la figure 3.19



**Figure 3. 19 Triangulation de Delaunay avec la portée de détection de chaque nœud**

Dans la triangulation de Delaunay le nombre de triangle qui est égal au nombre de cercle circonscrit de ces triangles dépasse généralement le nombre de nœud que nous désirons déployer dans un ROI comme le montre la figure 3.20.



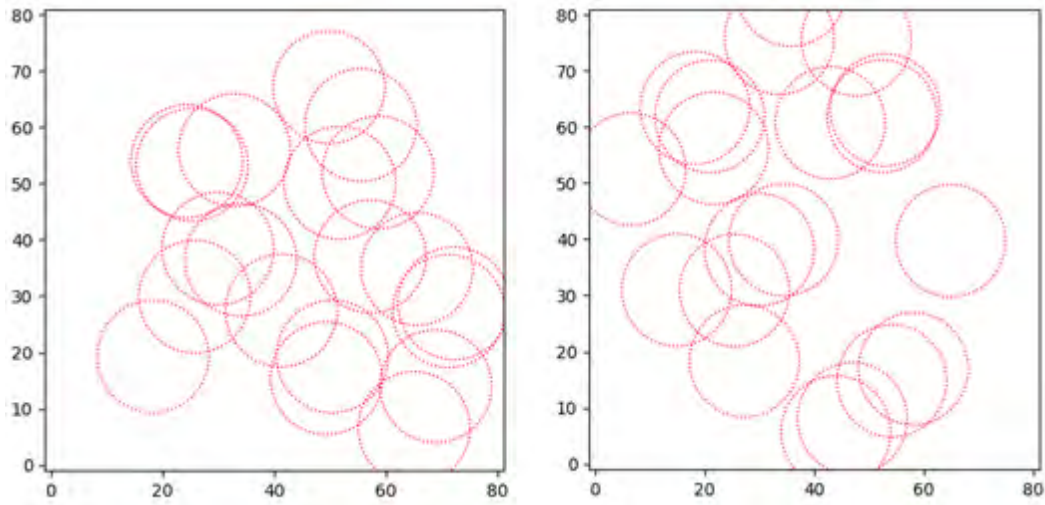
30 nœuds déployés par TD

29 nœuds déployés par TD

**Figure 3. 20 Dépassement de nombre des nœuds (2 solutions TD)**



Pour cela, nous avons pris les  $N$  premiers nœuds déployés pour chaque solution générée par TD, où  $N$  représente le nombre de nœuds que nous désirons déployer.



**Figure 3. 21 Déploiement initial de 20 nœuds par la TD (2 solutions)**

Notre déploiement initial (à la base des solutions TD) dans un champ d'intérêt  $80 \times 80$  est bien illustré sur les deux exemples de la figure 3.21.

La première étape de notre hybridation TDABC consiste à générer  $N$  solutions TD, chaque solution présente un déploiement initial des nœuds. Ce déploiement va être amélioré par l'algorithme d'optimisation ABC dont le but est de maximiser la couverture dans un champ d'intérêt.

### **3.3.1.2. Algorithme de Colonie des Abeilles Artificielles ABC**

Dans le scénario du réseau, on suppose que :

- les rayons de détection et de communication des capteurs sont tous les mêmes ( $r$ ),
- tous les capteurs ont la capacité de communiquer avec ces voisins à un saut.
- tous les capteurs sont mobiles.

Dans l'algorithme ABC, la position d'une source de nourriture représente une solution possible au problème d'optimisation et la quantité de nectar d'une source de nourriture correspond à la qualité (aptitude) de la solution associée. Donc le déploiement des capteurs dans la zone détectée (chaque solution du problème de déploiement) fait référence à une source de nourriture dans l'algorithme. Le taux de couverture du réseau, c'est-à-dire la zone de couverture totale, correspond à la valeur de fitness (nectar) de la solution. Le tableau 3.4 résume les paramètres

correspondants dans l'algorithme ABC par rapport à celle du déploiement des nœuds capteurs. Dans le modèle ABC des colonies d'abeilles artificielles, dans lesquelles le but des abeilles est de trouver la meilleure solution [194]. Le modèle comporte 3 groupes d'abeilles : les abeilles employées, les observateurs et les éclaireuses.

Une abeille qui est en attente dans la zone de danse pour déterminer le choix d'une source de nourriture, est un spectateur. Lorsqu'une abeille se rend à une source de nourriture précédemment visitée, c'est une abeille employée. Une abeille qui effectue des recherches aléatoires s'appelle une éclaireuse.

Les phases de l'algorithme ABC sont illustrées dans la figure 3.22.

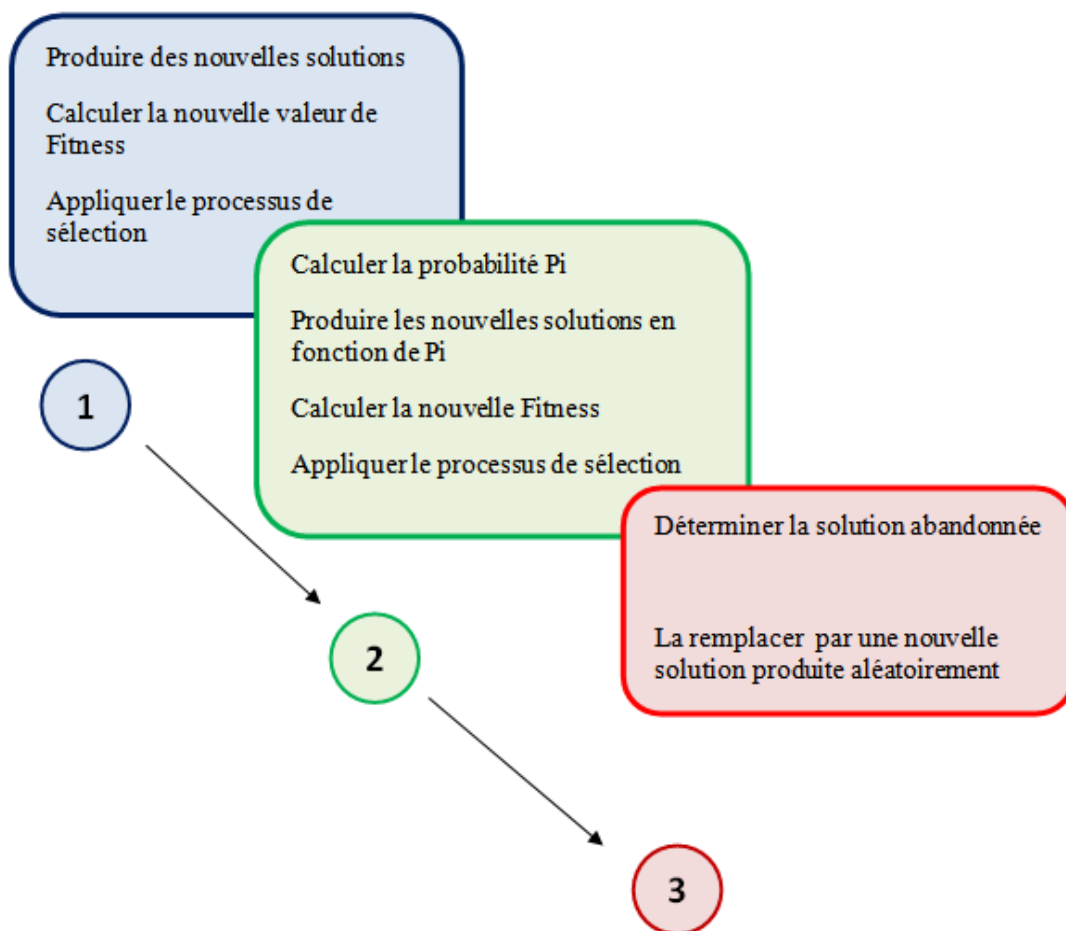


Figure 3. 22 Les phases de l'algorithme ABC

**Tableau 3. 4 Correspondance entre les paramètres de l'algorithme ABC, et du déploiement des nœuds capteurs**

Algorithme ABC	Problème de déploiement
Solution d'une source alimentaire	Solution d'un déploiement des nœuds
Dimensions N dans chaque solution	Coordonnées des N Nœuds capteurs
Fitness de la solution (quantité de nectar)	Taux de couverture de la zone d'intérêt
La source qui a plus de quantité de nectar	La solution qui a une répartition optimale des capteurs

Chaque solution représente un tableau qui a N objets (nœud capteur). La figure 3.23 montre un tableau qui présente une solution dans un espace 2D. Les éléments du tableau de solution sont les positions (x, y) des nœuds capteurs dans le réseau. La figure 3.24 montre un tableau qui présente une solution de déploiement dans un espace 3D.

x1	y1	x2	y2	x3	y3	x4	y4
16.21	08.65	19.75	10.30	07.39	12.60	18.36	03.71

**Figure 3. 23 Une solution de déploiement de quatre nœuds dans un espace 2D (20\*20)**

x1	y1	z1	x2	y2	z2	x3	y3	z3	x4	y4	z4
10.36	14.96	02.22	18.47	08.21	03.75	19.88	15.29	00.25	02.51	18.31	1.65

**Figure 3. 24 Une solution de déploiement de quatre nœuds dans un espace 3D (20\*20\*4)**

Chaque solution est présentée par :

$$S_i = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} \text{ où } i = 1, 2, \dots, \text{nb.}$$

nb et m représentent respectivement le nombre total des solutions (générées par le diagramme de Delaunay) et le nombre total de nœuds à déployer.



**3.3.1.2.1. Phase 0 : initialisation**

- Initialisation des paramètres : le rayon de détection  $r$  d'un nœud capteur, la taille de la zone d'intérêt  $A$ , le nombre de capteurs mobiles  $m$  à déployer dans le champ d'intérêt, la taille de la colonie  $cs$ , le nombre maximal d'itérations  $MaxCycle$  et la limite pour les abeilles éclaireuses  $l$ .
- Calcul du taux de couverture pour chaque solution  $x_i$  (chaque solution de Delaunay)
- Évaluation de la population: chaque source de nourriture  $x_i$  (chaque solution de Delaunay) sera évaluée.
- Initialisation du cycle  $c = 0$ .

**3.3.1.2.2. Phase 1: phase des abeilles employées**

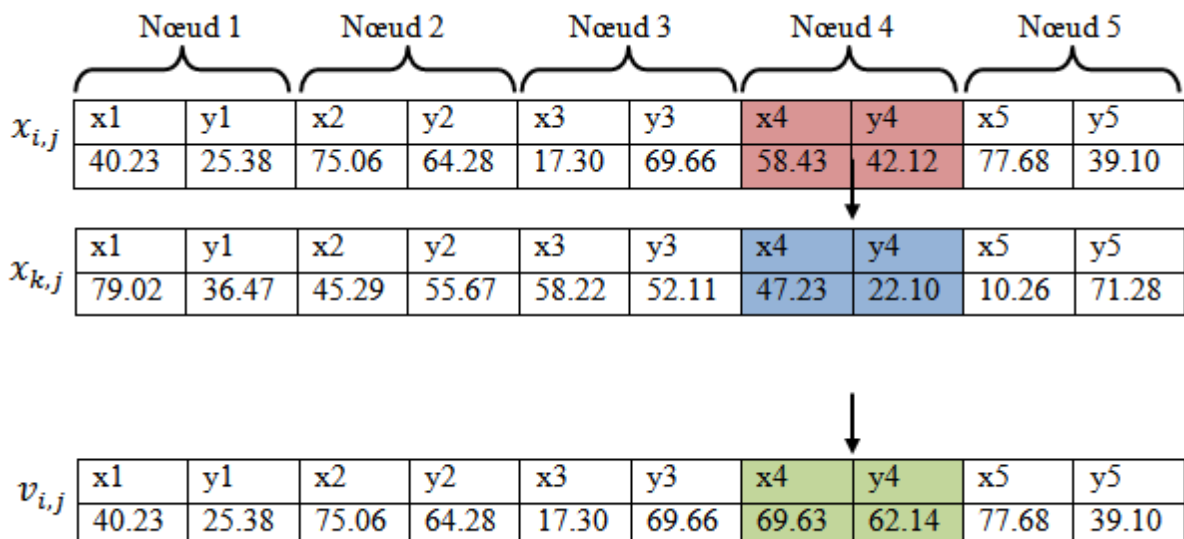
Soit  $i \in \{1,2,3 \dots S_i\}$  et  $j \in \{1,2,3 \dots D\}$

Les étapes suivantes seront répétées pour toutes les solutions :

- Produire de nouvelles solutions  $v_i$  au voisinage de  $x_i$  pour les abeilles employées en utilisant l'équation (16).

$$v_{i,j} = x_{i,j} + \varphi_{i,j} (x_{i,j} - x_{k,j}) \quad (16)$$

Avec :  $k$  qui est une solution au voisinage de  $i$ ,  $\varphi$  est un nombre aléatoire dans la plage  $[-1,1]$ , et  $j$  est la position du nœud capteur sélectionné au hasard. La figure 3.25 présente un exemple de la production d'une nouvelle solution en appliquant l'équation (16).



**Figure 3. 25 Produire une nouvelle solution**

- Vérifiez  $v_{i,j}$  pour rester dans les limites du champ d'intérêt.
- Calculer les valeurs de fitness des solutions, la formule suivante est utilisée

$$fitness_i = \begin{cases} \frac{1}{1+fitness_{(x_i)}} & si\ fitness_{x_i} \geq 0 \\ 1 + |fitness_{(x_i)}| & si\ fitness_{x_i} < 0 \end{cases} \quad (17)$$

- Appliquer le processus de sélection "gloutonne" (sélection de la source de nourriture qui a le plus de nectar) entre  $x_i$  et  $v_i$

**Tableau 3. 5 Exemple sur le processus de sélection gloutonne**

Solution	Fitness (couverture en %)
$x_i$	63.25%
$v_i$	65.87%

Dans l'exemple ci-dessus du tableau 3.5, le taux de couverture de la nouvelle solution  $v_i$  a dépassé le taux de couverture de la solution  $x_i$  dans ce cas la solution  $x_i$  va être remplacée par la solution  $v_i$ .

### **3.3.1.2.3. Phase 2: phase des abeilles spectatrices**

- Calculer les valeurs de probabilité  $p_i$  pour les solutions  $x_i$  au moyen de leurs valeurs de *fitness* en utilisant l'équation (18).

$$p_i = \frac{0.9 * fitness_i}{fitness_{best}} + 0.1 \quad (18)$$

**Tableau 3. 6 Exemple de calcul de probabilité d'une population de six solutions**

Solution	Probabilité
$x_1$	0.86
$x_2$	0.92
$x_3$	0.78
$x_4$	0.88
$x_5$	0.69
$x_6$	0.92

- Produire les nouvelles solutions  $v_i$  pour les abeilles observatrices à partir des solutions  $x_i$  sélectionnées en fonction de  $p_i$ . (choisir une solution en fonction de sa probabilité d'être choisie) et appliquer la même équation (16).
- Calculer les valeurs de fitness des solutions.
- Appliquer le processus de sélection gloutonne pour les spectateurs entre  $x_i$  et  $v_i$  :
  - Si la solution mutante  $v_i$  est meilleure que la solution actuelle  $x_i$ , remplacer la solution par le mutant et réinitialiser le compteur d'essai de la solution  $i$
  - Si la solution ne peut pas être améliorée, augmenter de son compteur d'essais (19) :
 
$$\text{Trial}[i] = \text{Trial}[i] + 1 \quad (19)$$
- Mémorisez la meilleure solution obtenue jusqu'à présent.

### 3.3.1.2.4. Phase 3 : phase des abeilles éclaireuses

L'algorithme 3-3 présente le pseudo code de la phase des abeilles éclaireuse, cette phase consiste à :

- Déterminer la solution abandonnée.
- S'il existe, la remplacer par une nouvelle solution produite aléatoirement en utilisant l'équation (16).

```

Void init (int index)
Début
    |
    | ...
    | fitness [index] = CalculFitness (f[index]);
    | trial [index] = 0;
end

Void SendScoutBees ()
Début
    |
    | ...
    | if (trial [maxtrialindex] >= limite) then
    | |   init (maxtrialindex);
    | end
end
    
```

**Algorithme 3- 3 Pseudo code de la phase des abeilles éclaireuses**

Dans la phase d'initialisation nous avons fixé le paramètre "limite pour les abeilles éclaireuses l", par la suite, nous pouvons déterminer la solution qui sera abandonnée :

après une "limite" du nombre d'essais, cette solution ne peut pas s'améliorer (taux de couverture), cette solution abandonnée sera remplacée par une nouvelle solution. L'exemple dans le tableau 3.7 montre que la solution 5 est abandonnée (car la limite = 10 est atteinte).

**Tableau 3. 7 Exemple de calcul de nombre d'essai (trial [i]) d'une population de six solutions**

Solution	Nombre d'essai
$x_1$	7
$x_2$	1
$x_3$	2
$x_4$	0
$x_5$	10
$x_6$	4

Enfin, il faut enregistrer la meilleure solution et répéter le processus jusqu'à la valeur maximale de C qui est le cycle maximal initialisé au début de l'algorithme.

### 3.3.2. Simulation

Pour les simulations, nous avons également utilisé OMNET++ pour évaluer notre algorithme TDABC dans une approche centralisée. Tout d'abord, nous avons appliqué le TD sur un ROI de 80\*80 mètres. TDABC commence son processus par l'exécution de TD qui va générer une population de 100 solutions (individus). Chaque solution présente les positions de chaque objet IoT dans un plan 2D. Les solutions données par TD représentent la population initiale de l'algorithme ABC. Le ABC permet d'optimiser les solutions trouvées par le TD en remplaçant les positions des nœuds afin de leur trouver le meilleur positionnement dans le ROI et d'avoir par la suite le degré de couverture maximisé. Le choix des valeurs des paramètres repose sur des tests empiriques. Le Tableau 3.8 montre les paramètres de simulation utilisés.

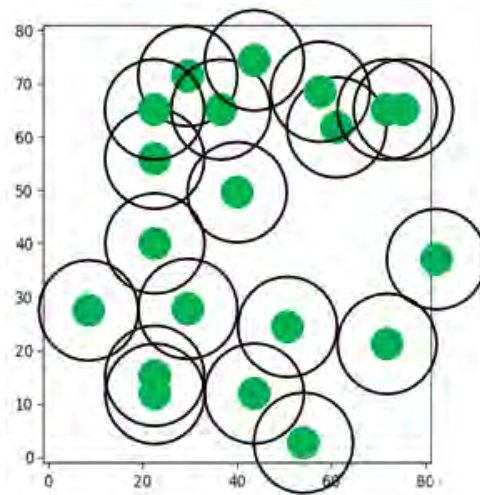
**Tableau 3. 8 Paramètres de simulation de TDABC**

Paramètres	
Taille de la population: Nombre des individus	100
Nombre d'itération	[100..1500]
Nombre des objets à déployer (nœuds)	20
Limite l " des abeilles éclaireuses "	30
Taille ROI	80*80
Condition d'arrêt	92 %
RSSI	100 (initial)
FER	0.01 (initial)
Portée de transmission intérieure	10 m
Portée de détection intérieure	10 m
Fréquence	2.4 GHz
Moyenne du nombre d'exécution	30
Antenne	isotropique
Température de fonctionnement du nœud	0° C à 60° C

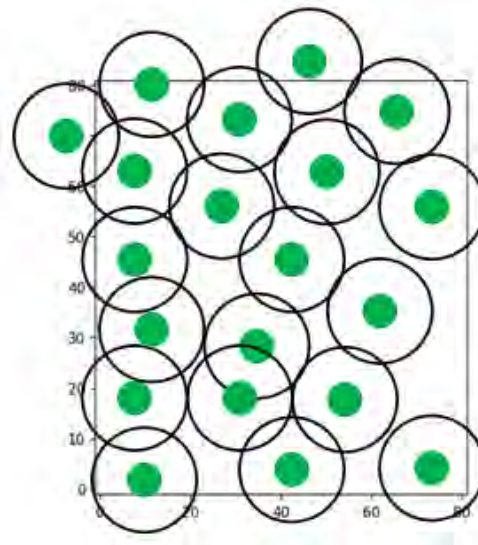
### 3.3.3. Résultats et discussions

Nous présentons dans cette partie les résultats obtenus de l'hybridation DVAG dans une approche centralisée avec des comparaisons en termes de taux de couverture, RSSI, durée de vie et nombre de voisins.

La figure. 3.26 et la figure. 3.27 montrent la différence, en termes de degré de couverture, entre la couverture aléatoire initiale générée uniquement par la Triangulation de Delaunay TD et le degré de couverture fourni par le TDABC proposé.

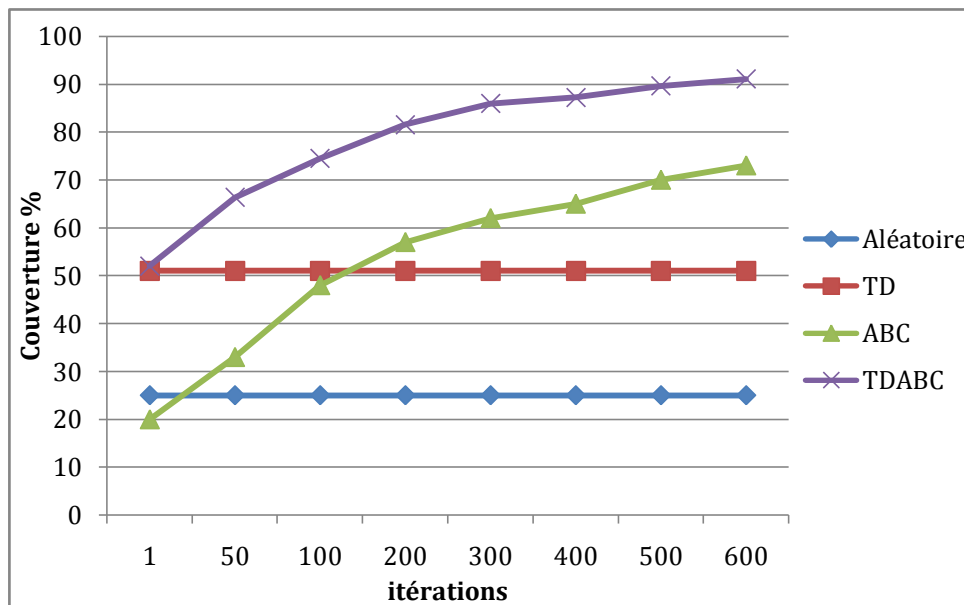


**Figure 3. 26 Déploiement aléatoire initial en utilisant TD**



**Figure 3. 27 Déploiement des nœuds après l'exécution de 650 itérations de TDABC**

La figure 3.28 montre un taux de couverture moyen de 30 exécutions du l'algorithme ABC, et en fonction du nombre d'itérations du ABC.



**Figure 3. 28 Comparaison du degré de couverture en fonction du nombre d'itérations pour TD, ABC et TDABC**

La figure 3.28 montre que le degré de couverture est amélioré lorsque le nombre d'itérations (cycles) ABC augmente. De plus, la figure 3.28 indique que le TD est

meilleur que l'algorithme ABC pour un nombre d'itérations d'ABC inférieur à 100, tandis que l'algorithme ABC donne un meilleur degré de couverture pour plus de 100 itérations. L'algorithme hybride TDABC est meilleur que TD et ABC dès la première itération de l'algorithme ABC.

Le tableau 3.9 présente les valeurs d'Hyper-volume (meilleur, moyen et pire) d'une 30 exécution de l'algorithme ABC, pour différentes valeurs d'objectifs et différentes valeurs d'itérations.

**Tableau 3. 9 Meilleure, moyenne et pire valeurs d'Hyper-volume de l'algorithme ABC**

Nombre d'objectifs	Nombre max d'itérations	ABC
1	100	0.9287 0.9103 0.8926
	400	0.9385 0.9269 0.9187
2	100	0.9223 0.9066 0.8875
	400	0.9359 0.9178 0.9025

Le tableau 3.9 montre qu'en considérant un seul objectif (le degré de couverture) au lieu de deux objectifs (degré de couverture et coût de déploiement), l'algorithme ABC donne des valeurs plus élevées (donc meilleures) d'Hyper-volume. De plus, augmenter le nombre d'itérations, permet à l'algorithme ABC d'obtenir de meilleurs résultats (valeurs plus élevées de l'Hyper-volume).

Le tableau 3.10 compare le comportement des algorithmes testés concernant les valeurs pour le RSSI, la durée de vie et le nombre moyen de voisins pour tous les nœuds (ces métriques sont définies précédemment).

**Tableau 3. 10 Comparaison des valeurs moyennes du RSSI, de la durée de vie et du nombre de voisins pour TD, ABC et TDABC**

	TD	ABC	TDABC
RSSI	130.38	132.28	136.12
Durée de vie (en secondes)	1258	1341	1732
Nombre des voisins	4.46	4.46	6.42

Le tableau 3.10 indique que l'algorithme hybride proposé est meilleur que l'algorithme ABC seul et le TD seul, pour le RSSI, la durée de vie et le nombre moyen de voisins. Une observation intéressante est que malgré les meilleures performances d'ABC par rapport au TD concernant la durée de vie et le nombre de voisins, ce dernier algorithme donne de meilleurs résultats que le ABC concernant les valeurs de RSSI.

Notre approche proposée montre une meilleure couverture par rapport à d'autres résultats [195]. Dans [195], une optimisation d'algorithme de colonie des abeilles artificielle modifié a été présentée avec différents scénarios et avec une couverture atteignant 90 %. Cependant, cette étude a utilisé un nombre élevé de nœuds mobiles et fixes pour parvenir à ces résultats (100 nœuds déployés dans un ROI de 100\*100). Notre approche proposée utilise moins de nœuds avec tout de même une meilleure couverture.

Afin de confirmer l'efficacité de notre algorithme, nous allons comparer la meilleure solution obtenue avec l'estimation du rapport d'approximation adaptative de TDABC, pour différents intervalles de temps, avec une complexité  $O(M \times P)^2$  où M est le nombre de nœuds mobiles et P est la taille de la colonie.

### **3.4. C-DVAG vs C-TDABC**

Dans cette partie nous allons comparer les deux hybridations centralisées C-DVAG et C-TDABC en termes de taux de couverture, coût temporel, RSSI et durée de vie.

#### **3.4.1. Taux de couverture**

Nous mesurons le pourcentage de couverture de surface dans les 2 hybridations, avec une condition d'arrêt d'un taux de couverture de 92%. Dans les deux hybridations, nous avons cumulé la moyenne des résultats trouvés qui atteint cette condition d'arrêt. Nous pouvons voir clairement que pour une approche centralisée l'hybridation DVAG nécessite plus que 200 itérations pour atteindre une couverture de la surface de 92%. DVAG atteint une couverture de 92.11% à l'itération 913 et 92.24% pour le TDABC à l'itération 708 comme le montre la figure 3.29.



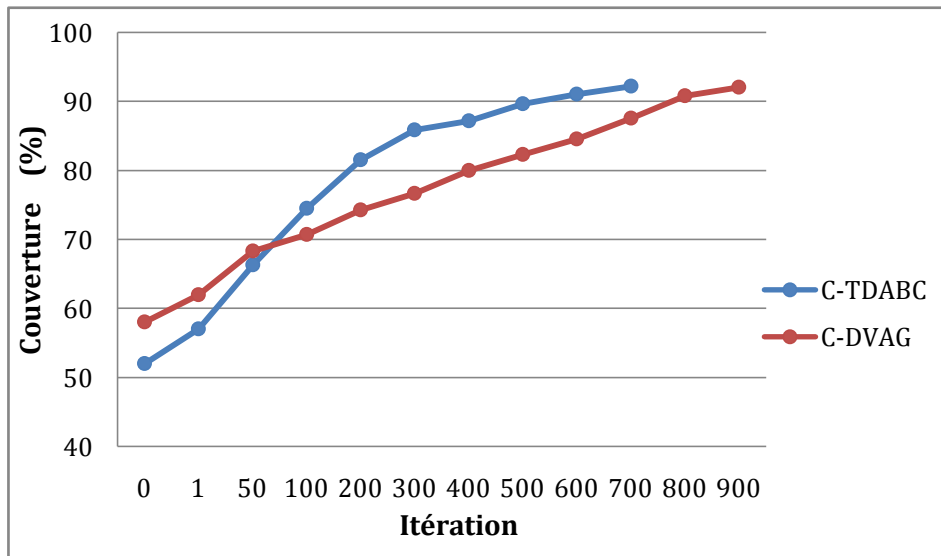


Figure 3. 29 Comparaison du degré de couverture entre C-DVAG et C-TDABC

### 3.4.2. Coût de temps de calcul

Nous avons cumulé la moyenne des résultats d'un déploiement de 10 nœuds pour calculer le coût de calcul des deux hybridations. L'hybridation centralisée C-TDABC a un coût de temps de calcul clairement élevé par rapport à l'hybridation C-DVAG, comme le montre la figure 3.30. Cela s'explique par le coût de calcul de chaque itération dans l'algorithme ABC, il y a un parcours de toutes les solutions dans les trois phases de cet algorithme avec le calcul de la fonction fitness pour chaque individu dans une grande population utilisée dans l'approche centralisée.

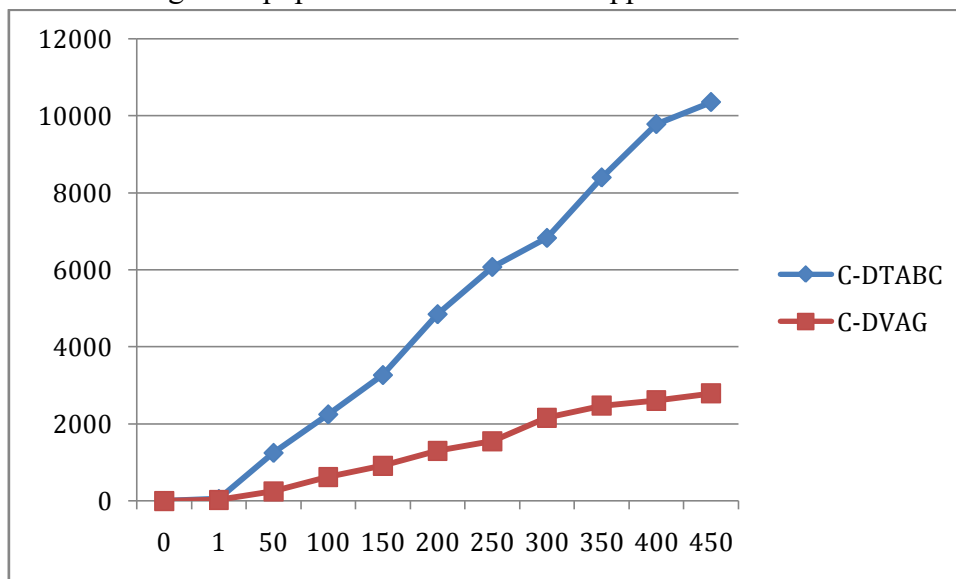


Figure 3. 30 Comparaison du coût de temps entre DVAG et TDABC

### 3.4.3. Calcul des valeurs RSSI, durées de vie et nombres de voisins

**Tableau 3. 11 Comparaison des valeurs moyennes des métriques réseau pour les différents algorithmes**

	DV	TD	AG	ABC	DVAG	TDABC
RSSI	132.39	130.38	131.83	132.28	137.28	136.12
Durée de vie (en secondes)	1298	1258	1367	1341	1693	1732
Nombres des voisins	4.23	4.46	4.65	4.46	5.98	6.42

D'après le tableau 3.11, on déduit que les algorithmes d'optimisations seules (AG et ABC) sont généralement plus performants que les méthodes géométriques (DV et TD) en termes de durée de vie et nombre de voisins. De plus, les hybridations DVAG et TDABC sont meilleures que les autres approches pour toutes les métriques évaluées. Par contre, on remarque que si on compare les méthodes d'une même classe (DV vs TD ; AG vs ABC), on trouve qu'il n'y a pas de supériorité totale d'une approche par rapport à une autre : par exemple, TD est meilleur que DV pour le RSSI mais DV donne un nombre de voisins meilleur que TD. De même, AG obtient un RSSI légèrement meilleur que ABC mais ce dernier donne un nombre de voisin meilleur que AG.

### 3.5. Conclusion

Pour résumer, nous avons dans ce chapitre décrit et présenté notre approche centralisée où l'exécution se fait dans un seul nœud. Nous avons utilisé pour résoudre le problème de déploiement optimisé dans les réseaux sans fils de collecte des données nos deux propositions d'hybridations DVAG et TDABC.

Une première hybridation DVAG entre le diagramme de Voronoi et l'algorithme génétique est suggérée. Le DV a généré au hasard les positions des objets IoT. Ces positions ont présenté la population initiale de l'AG. Ce dernier algorithme est utilisé pour améliorer les positions d'objets tout en maximisant le degré de couverture dans un champ d'intérêt. Par rapport à l'AG d'origine standard et au DV, l'algorithme hybride a montré une amélioration claire de la solution de déploiement. Une deuxième hybridation TDABC entre la Triangulation de Delaunay et l'algorithme de colonie des abeilles artificielles. Le TD génère une population initiale qui présente des solutions de déploiement. Ces solutions seront améliorées par l'algorithme de colonie des abeilles artificielles dans le but de maximiser le taux de couverture d'un champ d'intérêt.

Pour une approche centralisée, l'hybridation DVAG nécessite plus que 200 itérations par rapport à TDABC pour atteindre une couverture de la surface de 92%. Tandis que l'hybridation TDABC a un coût de temps de calcul clairement élevé par rapport à l'hybridation DVAG.

Dans le chapitre suivant, nous nous attacherons à présenter l'approche maintenant distribuée pour les deux hybridations DVAG et TDABC. Nous présentons par la suite les résultats d'évaluations de DVAG et de TDABC pour les deux approches sur un environnement expérimental en prototypant sur un banc d'essai réel en utilisant des nœuds M5StickC équipés d'un processeur ESP32.

## **4. Approche Distribuée : protocole de communication, simulations, expérimentations et résultats**



## 4.1. Introduction

Dans le chapitre précédent, nous avons présenté une approche centralisée de deux hybridations DVAG et TDABC où l'exécution se fait dans un seul nœud. Nous proposons dans ce chapitre une approche distribuée pour DVAG et TDABC où le traitement se fait dans plusieurs nœuds en parallèle. L'objectif est d'atteindre la solution souhaitée dans des délais plus courts, avec une meilleure performance par rapport à une approche centralisée. Cette approche vise le positionnement des nœuds dans un réseau et par conséquent la maximisation de la couverture dans un ROI avec le minimum des nœuds.

Pour ce qui est de la première hybridation DVAG, le diagramme de Voronoï DV présente le point de départ de notre travail qui générera une population initiale présentant un déploiement aléatoire d'objets. Cette génération sera exécutée sur un seul nœud et la population sera distribuée protocolairement sur l'ensemble des autres nœuds formant le réseau de collecte des données. Un algorithme génétique est exécuté dans chacun de ces nœuds dans le but d'optimiser ce déploiement. De même, l'hybridation TDABC commence par la génération de la population initiale à travers la Triangulation de Delaunay TD. Cette génération sera exécutée dans un seul nœud et la population générée sera distribuée protocolairement sur l'ensemble des autres nœuds formant le réseau de collecte des données RCD. Ceci fait que l'algorithme ABC est exécuté en parallèle dans chacun de ces nœuds.

Dans le cas où on a un déploiement existant initial, la phase de positionnement avec le DV ou le TD sera inutile. Nous pouvons remplacer cette dernière phase par une phase initiale dans laquelle les nœuds doivent connaître les positions des voisins. La solution proposée est de faire une diffusion où chaque nœud diffuse sa position aux voisins à portée. Chaque nœud connaît sa position par un protocole de localisation [202,203].

L'approche proposée est évaluée pour les deux hybridations sur un environnement expérimental par prototypage sur un banc d'essai réel à l'aide de nœuds M5StickC équipés d'un processeur ESP32. Nous verrons que les résultats révèlent qu'une approche distribuée dépasse celles des algorithmes originaux en termes de degré de couverture, RSSI, durée de vie et nombre de voisins d'objets. Cette approche dépasse aussi une approche centralisée en termes de rapidité de calcul.

Ce chapitre est réparti en cinq sections : la section 2 traite l'approche distribuée pour notre première hybridation DVAG où nous présentons la stratégie de la distribution, le protocole de communication entre les nœuds ainsi que les résultats obtenus. Dans la section 3, nous traitons l'approche distribuée pour TDABC. La section 4 compare les deux hybridations DVAG et TDABC dans l'approche distribuée. Nous nous attachons à la fin de ce chapitre à comparer les performances atteintes entre les résultats obtenus par l'approche centralisée (chapitre 3) face aux avec les résultats obtenus par l'approche distribuée.

## 4.2. Approche distribuée pour DVAG

Cette contribution DVAG- distribuée fait l'objet de notre publication [198].

### 4.2.1. Stratégie de la distribution

Dans notre approche proposée illustrée en figure 4.1, il existe deux types de nœuds, un nœud Voronoi (Nœud.V) qui a pour but de générer une population initiale (N solutions Voronoi) qui va être subdivisée en plusieurs sous-populations. Chaque sous-population sera envoyée à un nœud génétique (Nœud.G). L'hybridation entre les deux algorithmes se fait principalement au niveau de chaque Nœud.G. Les valeurs de fitness relatifs aux individus sont calculées dans chacun des Nœud.G. dans le but est d'améliorer la couverture. Les solutions optimales trouvées seront renvoyées au Nœud.V.

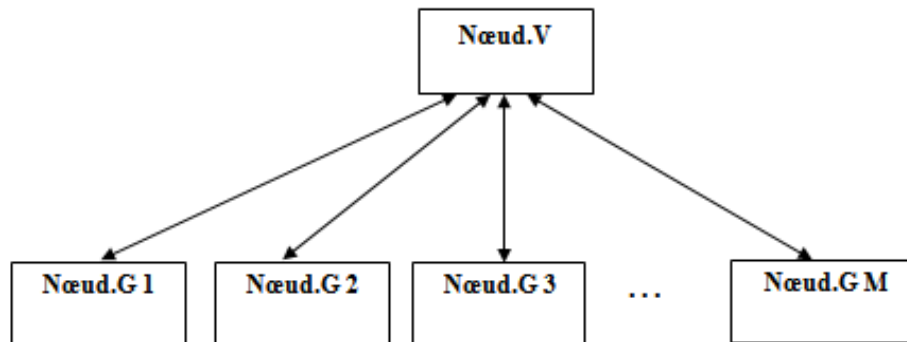


Figure 4. 1 Architecture distribuée proposée pour DVAG

L'idée de base est de développer un protocole permettant de gérer et de réaliser la communication entre les différents nœuds dans l'approche que nous avons proposée dans la figure 4.1.

### 4.2.2. Protocole de communication

Notre protocole est à la base de protocole "ESP-NOW" qui dispose d'une bibliothèque dédiée sous Arduino. Selon [199] ESP-NOW est un protocole de communication développé par Espressif, qui permet à plusieurs cartes électroniques de communiquer entre elles sans utiliser le WiFi standard, par exemple basé sur un point d'accès en mode infrastructure. ESP-NOW se base sur les liens directs entre les nœuds formant un réseau ad-hoc. Ce protocole est similaire à une connectivité sans fil à haute vitesse permettant l'échange des petits paquets de données à travers des bandes de fréquence 2.4GHz avec une basse consommation. Sa seule contrainte est que tous les nœuds doivent être à portée radio de tous les autres, car il n'y a pas de routage multi-saut prévu. Ceci n'est pas gênant pour notre expérimentation

d'intérieur. Pour aller plus loin, nous pourrions utiliser d'autres protocoles comme PainlesMesh [201]...

Afin de développer notre protocole et de réaliser l'approche distribuée, nous utilisons le même type des nœuds «M5StickC ESP32-PICO Mini IoT Development Kit » [200].

L'idée consiste à diviser la population initiale en sous-populations plus petites et exécuter simultanément les tâches correspondantes sur différents nœuds. Par exemple une population de 300 solutions générées par le Nœud.V est répartie sur 10 Nœud.G. Chaque nœud Nœud.G va travailler une sous-population de 30 individus.

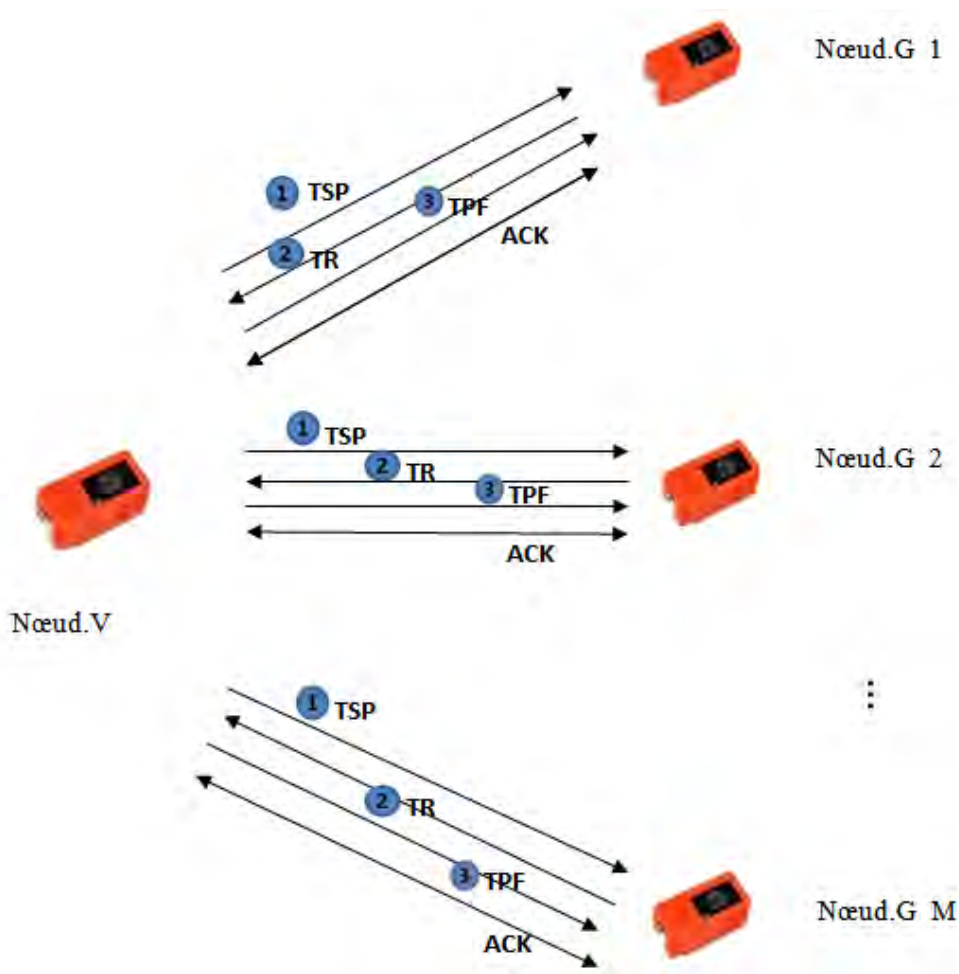
Notre protocole est basé sur quatre types de trames où chaque trame a une charge utile allant jusqu'à 250 octets transportable :

- TSP : une Trame-Sous-Population qui contient M solutions Voronoi, ce type de trame sera envoyé du Nœud.V vers tous les Nœud.G.
- TR : Une Trame-Résultat qui contient la meilleure solution trouvée pour chaque Nœud. Chaque Nœud.G renvoie une trame TR au Nœud.V après l'exécution de son algorithme.
- TPF : Une Trame-Position-Finale contenant la position finale de chaque nœud, le Nœud.V rassemble toutes les trames TR renvoyées par chaque Nœud.G et prendra par la suite la décision de la meilleure solution qui présente la couverture maximale trouvée.
- ACK : ou Accusé de réception, transmis dans les deux sens (figure 4.3), est utilisé pour informer la couche protocolaire du succès ou de l'échec de la transmission de chaque type de trame.

La quantité d'information la plus grande est au niveau de la trame TSP, nous pouvons donc parler d'une segmentation équivalente au niveau de la couche TCP/IP, c'est-à-dire que la trame TSP sera subdivisée en plusieurs sous-trames.

La figure 4.2 schématise notre architecture réseau proposée pour élaborer l'approche distribuée de DVAG





**Figure 4. 2 Architecture de réseau proposée**

L'un des avantages de notre protocole est d'avoir la capacité technique de s'adapter à des situations difficiles voire exceptionnelles : si soudainement, l'une des appareils se réinitialise ou dans le cas où il ya une perte d'alimentation, lorsque le nœud redémarre, il se "connectera" automatiquement à son homologue pour poursuivre la communication.

La Figure.4.3 présente un diagramme de séquence de notre protocole de communication appliqué sur un exemple de trois nœuds, chaque nœud a un rôle de transmission et de réception (T/R) des trames des données. Le Nœud.V au milieu et les deux Nœud.G sont à gauche et à droite. A l'instant  $t=0$  le Nœud.V commence par la génération des solutions Voronoi puis va répartir ces solutions aux différents Nœud.G. Dans cet exemple, les solutions Voronoi générées sont réparties sur 6 trames TSP, les 3 premières trames pour Nœud.G n° 1 et les 3 deuxièmes trames pour Nœud.G n°2, le délai d'envois de chaque trame est de 10ms. Chaque Nœud.G reçoit ses trames TSP (le nombre de trames reçues représente la taille de sous-population) qui présentent la population initiale pour exécuter son algorithme DVGA. A la suite de l'exécution, chaque Nœud.G envoie la solution trouvée dans une trame TR, le

Nœud.V qui est toujours « en écoute », rassemble toutes les trames TR et prend ainsi la décision de meilleure solution trouvée c'est à dire la solution qui présente la meilleure couverture. Cette solution sera diffusée sur les Nœud.G par la trame TPF.

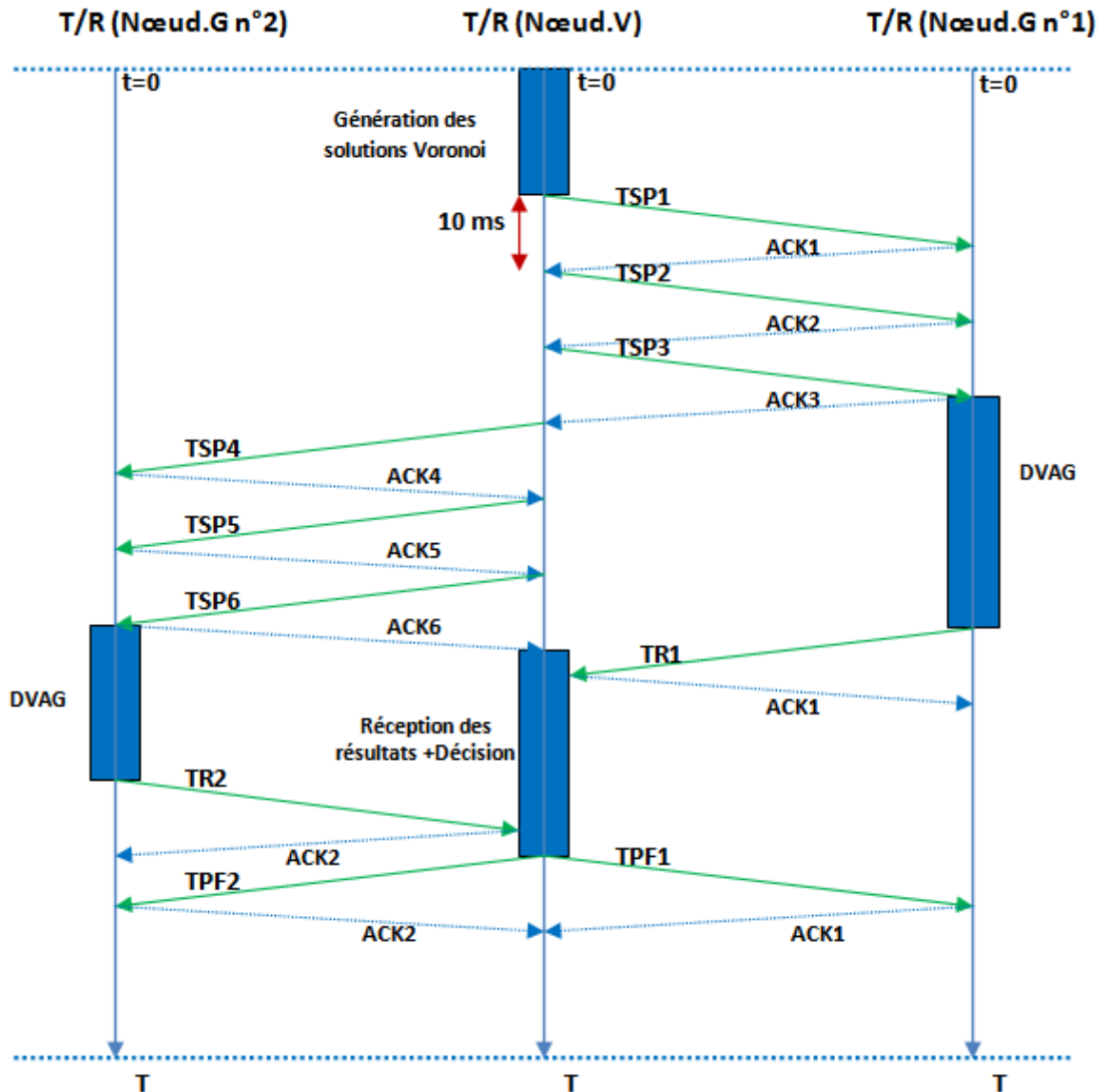


Figure 4. 3 Diagramme de séquence de notre protocole de communication entre trois nœuds

### 4.2.3. Conception de l'algorithme de positionnement distribué

#### 4.2.3.1. Scénario de déploiement

Le fonctionnement de notre approche comme la montre la figure.4.4 commence par la génération des solutions Voronoi dans le Nœud.V, ce même nœud subdivise la

population générée en sous-populations. Chaque sous-population sera affectée à un Nœud.G. Le processus développe indépendamment sa population jusqu'à ce qu'il décide de rassembler ses meilleurs individus qui seront candidats à l'étape de décision.

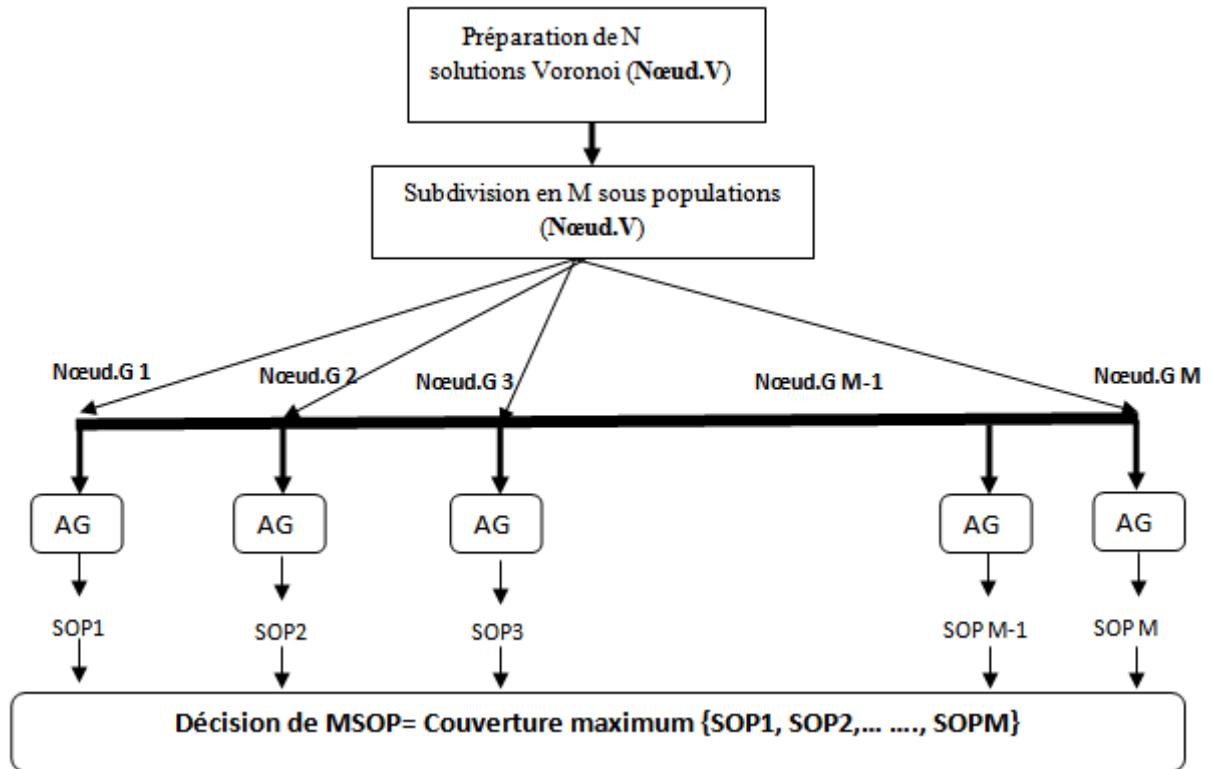


Figure 4. 4 Fonctionnement de l'approche proposée

#### 4.2.3.2. Algorithme d'optimisation hybride

L'algorithme 4-1 illustre le pseudo-code de notre algorithme développé afin d'élaborer l'approche distribuée de DVAG. Cet algorithme commence par l'implémentation d'un Diagramme de Voronoi pour générer une population initiale de solutions Voronoi (Popinitial) puis la subdivise en sous-populations (lignes 1 et 2). Ensuite, l'algorithme génétique est exécuté en parallèle dans tous les Nœud.G jusqu'à ce que la couverture maximale souhaitée soit obtenue (lignes 3 et 4 de l'algorithme). Les entrées de l'algorithme sont :

- Le nombre de nœuds à déployer M,
- Le nombre de solutions à générer par le Nœud.V, N-solutions représentent la population initiale,
- Les dimensions de la région d'intérêt (RoI),
- La taille de la sous-population pop\_size,

- La condition d'arrêt (soit par un taux de couverture cherché soit par un nombre d'itération définis dès le début).

**0) Définir la fonction d'évaluation f (Couverture maximale cherchée) ;**  
**1)  $Pop_{initial} \leftarrow$  Générer par le diagramme de Voronoi : N solutions ;**

**2) Subdiviser la population initiale (Pop) en sous-population de taille (pop\_size)**  
**3) Faire en parallèle sur chaque Nœud.G**

**3.1) Réception des solutions Voronoï**  
**3.2) calcul de taux de couverture pour chaque solution Voronoï**  
**3.3) Appliquer l'AG sur chaque Nœud.G (M solutions de Voronoi pour chaque nœud):**  
**Répéter**

**a) Sélection**  
**b) Croisement**  
**c) Mutation**

**Jusqu'à la couverture cherchée**

**3.4) Chercher la meilleure solution parmi toutes les solutions générées par chaque nœud**  
**3.5) Aller à 3 ou Stopper,**

**4) Fin du parallélisme**  
**5) Renvoyer la meilleure Solution.**  
**6) Affectation de position Finale pour chaque nœud**

#### **Algorithme 4- 1 Pseudo-code de DVAG distribuée**

Le fonctionnement de l'algorithme de Diagramme de Voronoi et l'algorithme génétique ont déjà été présentés à la section 2 de chapitre 3.

#### **4.2.4. Résultats et discussions**

Pour évaluer notre approche distribuée de déploiement des nœuds en appliquant notre algorithme DVAG dans un champ d'intérêt, un véritable prototype a été proposé, il a été testé dans un environnement réel avec des expérimentations réalisées sur un banc d'essai contenant de 6 à 10 nœuds. Avant ce prototypage, nous avons débuté par des simulations.

##### **4.2.4.1. Simulation**

Pour l'approche distribuée de l'algorithme DVAG, nous avons utilisé le même simulateur que l'approche centralisée présentée au chapitre 3, OMNET++ pour les simulations. Le choix des valeurs des paramètres, repose sur des tests empiriques.

Le processus de la distribution se fait dans les deux types de nœuds comme suit :

- Pour le Nœud.V : nous appliquons le DV sur un ROI de 80\*80 mètres, le DV divise la région en un ensemble de cellules de Voronoi pour générer une population de 100 solutions (individus) qui présentent les positions de chaque objet IoT dans un plan 2D, ensuite cette population est répartie en des sous-populations sur 20 nœuds. Le tableau 4.1 montre les paramètres de simulation utilisée pour le Nœud.V.

**Tableau 4. 1 Paramètres de simulation de D-DVAG pour le Nœud.V**

<b>Paramètres</b>	
Taille de la population: Nombre des solutions Voronoi à générer	100..200
Taille ROI	80*80
Nombre des objets à déployer (déploiement initiale des nœuds)	20
Taille de chaque sous- population	10..20
RSSI	100 (initial)
FER	0.01(initial)
Portée de détection intérieure	10 m
Fréquence	2.4 GHz
Moyenne du nombre d'exécution	30
Antenne	isotropique
Température de fonctionnement de nœud	0° C to 60° C

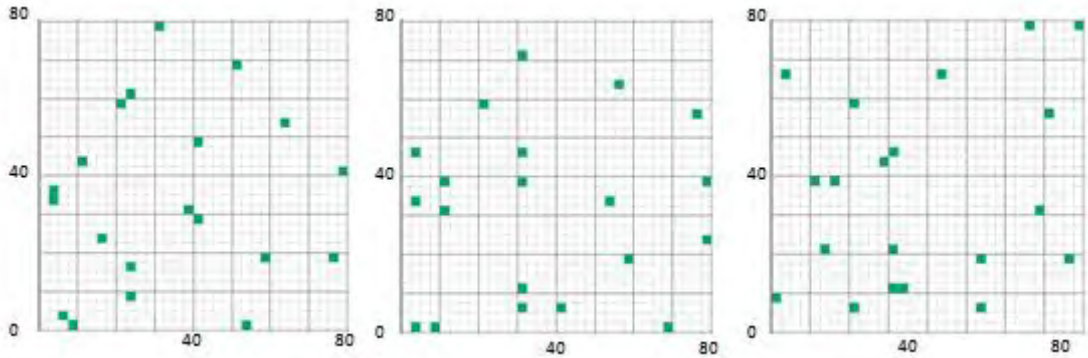
- Pour le Nœud.G : nous exécutons l'algorithme AG (après l'hybridation avec le DV) sur chacun des Nœud.G en parallèle sur le même ROI de 80\*80. L'AG permet d'optimiser les solutions trouvées par le DV en remplaçant les positions des nœuds afin de trouver le meilleur emplacement dans le ROI et d'avoir par la suite le degré de couverture maximal cherché. Le tableau 4.2 montre les paramètres de simulation utilisée pour chaque Nœud.G.

**Tableau 4. 2 Paramètres de simulation de D-DVAG pour chaque Nœud.G**

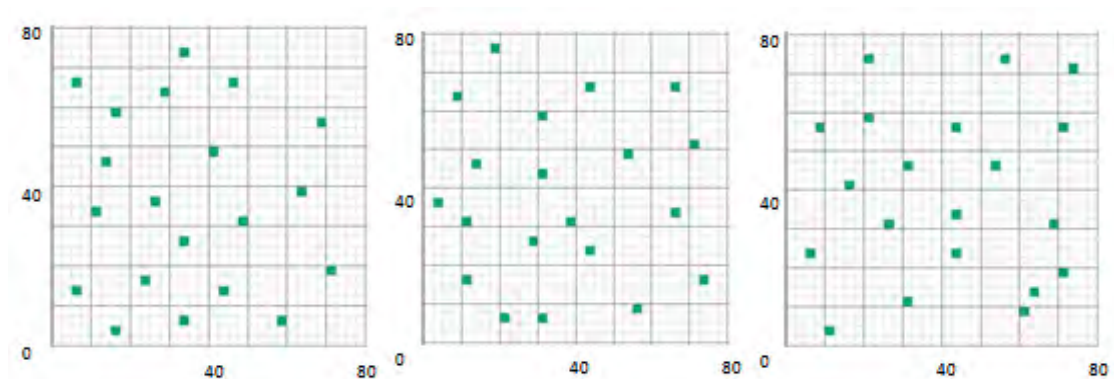
<b>Paramètres</b>	
Taille de la population reçue	10..20
Nombre d'itération	[100..1300]
Nombre des objets à déployer (nœuds)	20
Taux de croisement	0.9
Taux de mutation	0.1
Dimension ROI	80*80
Portée de détection d'un nœud	10
Condition d'arrêt (taux de couverture)	94%

RSSI	100 (initial)
FER	0.01(initial)
Fréquence	2.4 GHz
Moyenne du nombre d'exécution	30
Antenne	isotropique
Température de fonctionnement du nœud	0° C à 60° C

La figure 4.5 et la figure 4.6 montrent clairement la différence, en termes de degré de couverture, entre la couverture aléatoire initiale générée uniquement par le diagramme de Voronoï et le degré de couverture fourni par le DVAG proposé.



**Figure 4. 5 Déploiement de 20 nœuds avec DV seulement (exemple de 3 solutions)**



(a) Solution 1

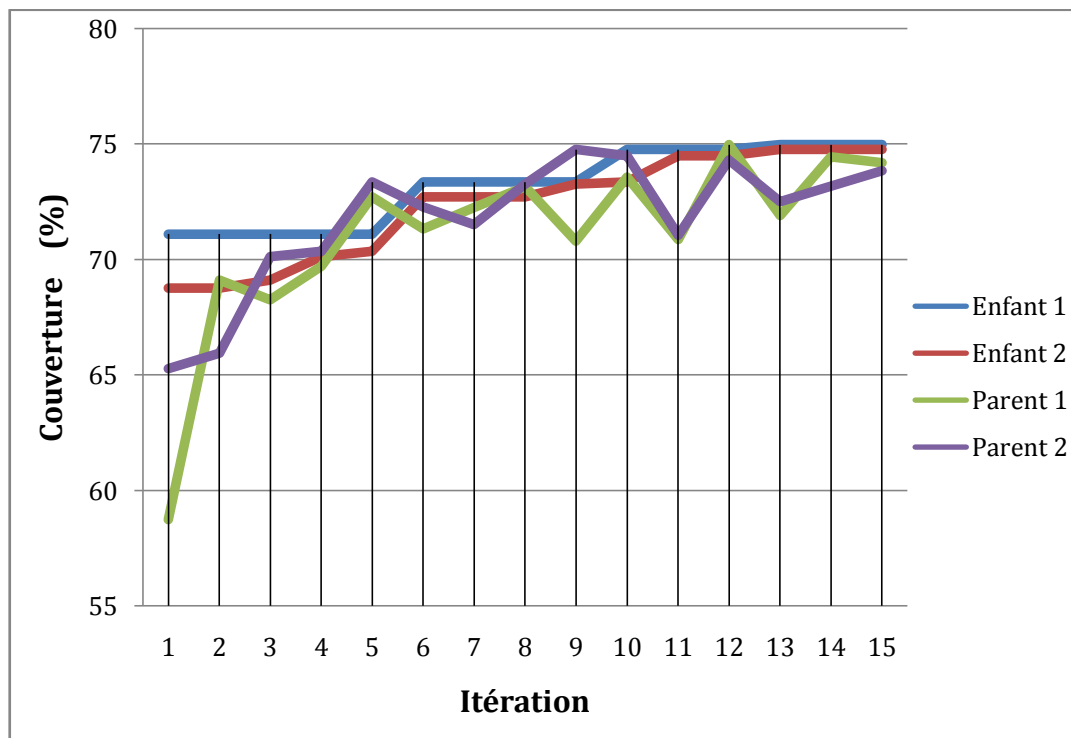
(b) Solution 2

(c) Solution 3

**Figure 4. 6 Déploiement de 20 nœuds après l'exécution de D-DVAG (exemple de 3 solutions).**

La figure 4.7 montre l'augmentation du degré de couverture après chaque itération de D-DVAG.





**Figure 4. 8 Fonctionnement d'AG dans chaque Nœud.G**

La figure 4.9 présente le degré de couverture pour six nœuds utilisés dans une approche distribuée, chaque nœud atteint sa couverture désirée (94%) dans un nombre d'itération différent d'un autre nœud. Par exemple, le nœud 4 a trouvé sa couverture à dans l'itération 706, tandis que le nœud 5 a trouvé sa couverture à l'itération 952.



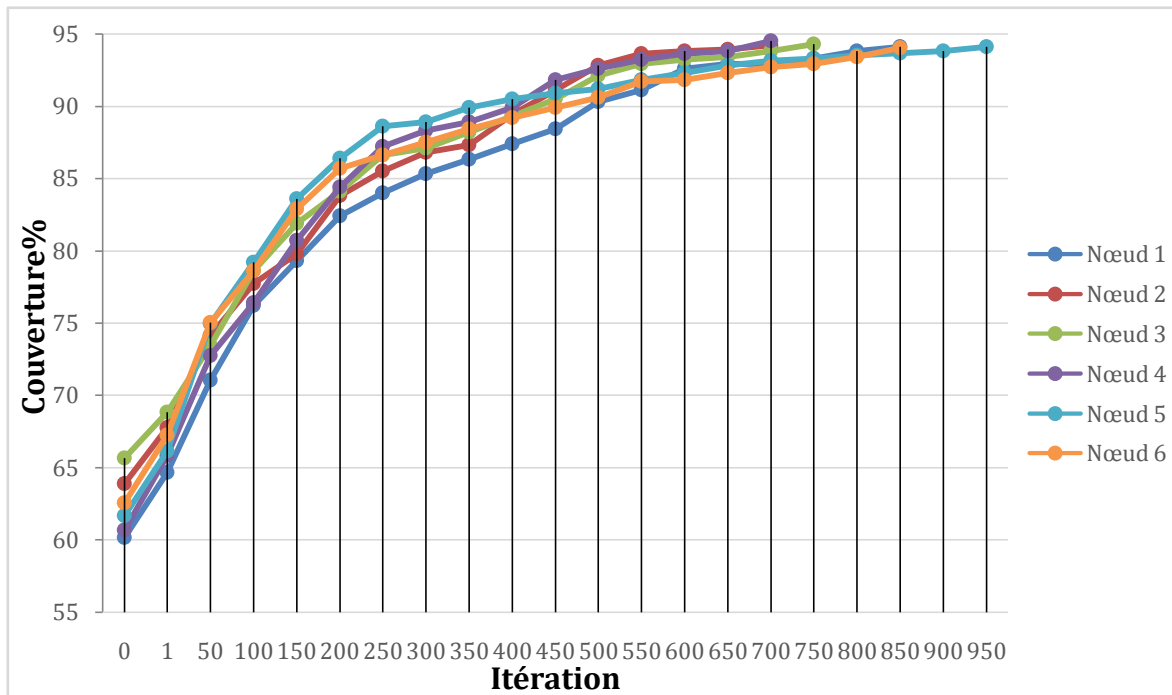
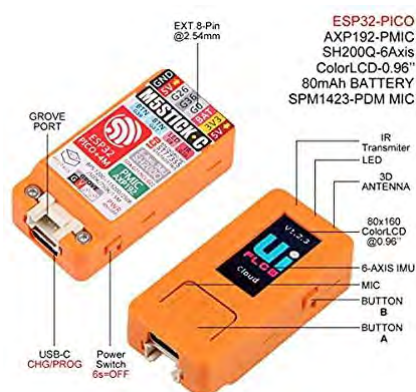


Figure 4. 9 Degré de couverture dans l'approche distribuée ayant six nœuds

L'intérêt d'avoir fait ses simulations avec OMNET++ est de comprendre l'efficacité de l'algorithme d'optimisation, en intégrant les échanges de trames entre les nœuds

#### 4.2.4.2. Expérimentations

Un véritable banc d'essai mettant en œuvre plusieurs nœuds réels a été conçu et organisé dans un réseau sans fil. Un seul type de nœud est utilisé dans nos expérimentations. Il s'agit d'un nœud M5StickC de la famille M5Stack animé par un processeur ESP32. Les nœuds M5StickC sont des cartes électroniques équipées d'une mémoire Flash de 4 Mo, d'une antenne 2.4 GHz, d'un émetteur IR, d'un microphone, boutons, d'un écran LCD (0.96 pouces), et d'une batterie interne LiPo, comme le montre la figure 4.10. Les nœuds M5StickC peuvent être développés avec les plateformes UIFlow, MicroPython et Arduino C.



**Figure 4. 10 Les nœuds M5StickC ESP32 utilisés [200]**

Le tableau 4.3 présente les paramètres d'expérimentation utilisés.

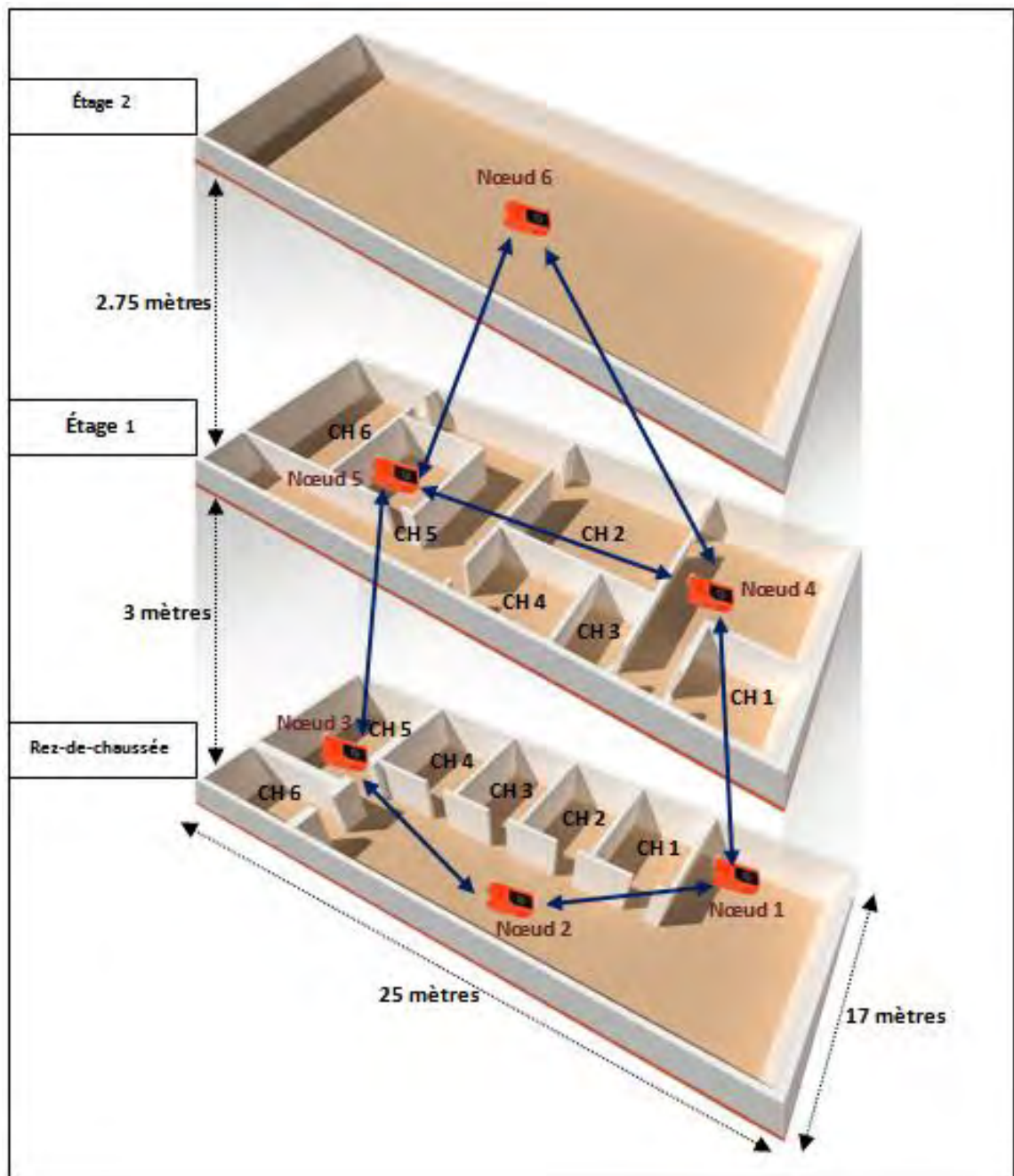
**Tableau 4. 3 Les paramètres d'expérimentation de DVAG**

Paramètres	
Taille de la population	100
Nombre d'itération	[700..1000]
Nombre des objets à déployer (nœuds)	06
Taux de croisement	0.9
Taux de mutation	0.1
Dimension de ROI	25*12 (RDC+2 étages)
Condition d'arrêt (taux de couverture)	94%
Seuil de résultats reçus (en %)	80%
RSSI	100 (initial)
FER	0.01(initial)
Portée de détection intérieure	10 m
Fréquence	2.4 GHz
Moyenne du nombre d'exécution	30
Antenne	isotropique
Température de fonctionnement du nœud	0° C à 60° C

Le cadre des expérimentations que nous avons utilisé est le suivant : 6 nœuds M5StickC sont déployés dans un appartement composé d'un rez-de-chaussée et de 2 étages, chacun avec une surface de 25 x 17 mètres. Le rez-de-chaussée et le premier étage contient chacun six chambres, une séparation murale de 20 à 30 cm d'épaisseur

entre 2 chambres voisines comme ils sont illustré dans la figure 4. 11, un plafond béton de 30 cm entre les deux étages. Le deuxième étage est un espace libre. Nous avons mesuré la portée de transmission des nœuds dans cet environnement, qui se situe est dans un intervalle allant de 10 à 18 m selon les obstacles rencontrés lors de la transmission. Les taux de RSSI (Received Signal Strength Indicator) et FER (Frame Error Rate) sont également calculés. La puissance de transmission du WiFi est de 100 mW pour respecter la législation. En raison de la nature stochastique des algorithmes d'optimisation, le processus d'optimisation est exécuté 30 fois et une valeur moyenne est calculée pour toutes les valeurs des résultats présentés dans les différentes figures.

Nous constatons que la plupart des Nœud.G trouvent leurs solutions optimales dans un délai similaire, alors que des autres prennent plus de temps pour la récupération des données et pour la prise de décision réalisant la meilleure solution. Comme nous définissons notre condition d'arrêt par une couverture désirée, nous avons besoin d'ajouter un nouveau paramètre "Seuil de résultat reçu". Par exemple si nous disposons de 10 nœuds à déployer et si nous aboutissons à un seuil de 80% des résultats, le Nœud.V une fois qu'il a atteint les résultats de 8 Nœud.G, il oblige le processus à s'arrêter afin de prendre une nouvelle décision.



**Figure 4. 11 Environnement d'expérimentation de DVAG**

### Calcul des valeurs RSSI, durées de vie et nombres de voisins

Les tableaux 4.4, 4.5 et 4.6 représentent les valeurs moyennes du RSSI, de la durée de vie et du nombre de voisins des nœuds utilisés.

**Tableau 4. 4 Valeurs moyennes du RSSI pour 6 nœuds**

Noeud	DV	AG	DVAG
1	132.39	131.83	137.28
2	135.28	128.92	136.64
3	131.23	125.17	131.55
4	114.72	116.28	115.21
5	89.29	92.88	91.53
6	129.82	127.76	129.16

Le tableau 4.4 montre des valeurs élevées de RSSI issues des nœuds positionnés selon l'algorithme DVAG. La supériorité de ces valeurs RSSI pour l'hybridation DVAG par rapport à DV et à AG indique une meilleure capacité de localisation des nœuds à l'aide de cet algorithme, si un algorithme de localisation se basait sur des mesures de distance entre les objets connectés par ce biais.

**Tableau 4. 5 Valeurs moyennes de durée de vie (en secondes) pour 6 nœuds**

Noeud	DV	AG	DVAG
1	1298	1367	1693
2	987	927	996
3	1029	1035	1083
4	703	749	753
5	870	883	892
6	943	932	921

La durée de vie du réseau est calculée selon la durée dans lequel le premier nœud est hors service suite à l'épuisement de son batterie. La durée de vie affecte sur la qualité de solution de déploiement cherchée.

Le tableau 4.5 illustre les durées de vie élevées des nœuds positionnés selon l'algorithme DVAG. Cela indique une meilleure capacité de couverture (déploiement) des nœuds à l'aide de cet algorithme.

**Tableau 4. 6 Valeurs moyennes du nombre de voisins pour 6 nœuds**

Node	DV	AG	DVAG
1	4.23	4.65	4.98
2	3.22	3.41	3.56

3	2.68	2.91	3.02
4	3.56	3.86	3.94
5	4.29	4.44	4.62
6	3.12	3.42	3.68

Le tableau 4.6 indique des nombres élevés de voisins des nœuds déployés selon l'hybridation DVAG. La supériorité de ce nombre de voisins reflète la meilleure qualité de connectivité des nœuds utilisant dans cette hybridation.

### 4.3. Approche distribuée de D-TDABC

#### 4.3.1. Stratégie de distribution

Nous allons utiliser la même stratégie de distribution de DVAG. Dans l'approche distribuée de TDABC, nous utilisons 2 types de nœuds comme le montre la figure 4.12:

- Un nœud Delaunay nommé Nœud.TD qui a pour but de générer des solutions de Delaunay, ces solutions représentent la population initiale. Cette population sera subdivisée en des sous-populations.
- Un nœud ABC nommé Nœud.ABC qui va recevoir une sous-population de solutions de Delaunay. Cette sous-population présente le point de départ pour exécuter l'algorithme ABC.

L'hybridation entre les deux algorithmes de TDABC se fait principalement au niveau de chaque Nœud.ABC. Les valeurs de fitness des individus sont calculées dans chacun des Nœud.ABC dont le but d'améliorer la couverture. Les solutions optimales trouvées seront alors renvoyées au Nœud.TD. Ainsi l'approche est basée sur une communication bidirectionnelle.

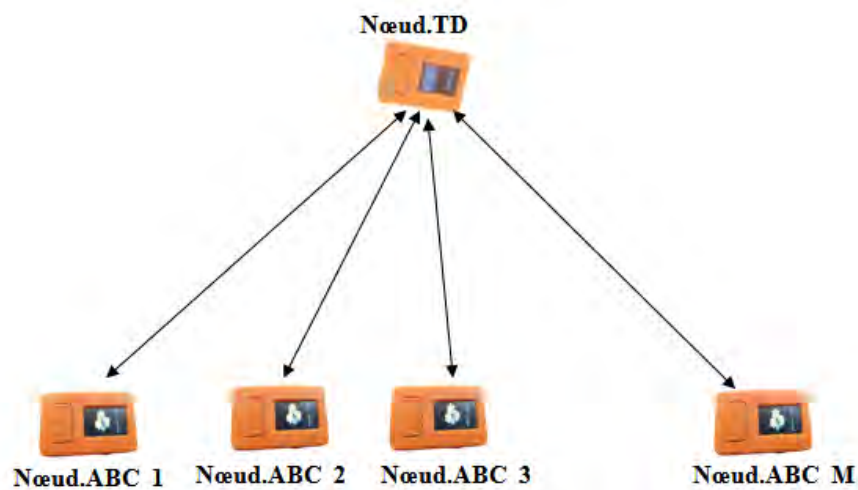


Figure 4. 12 Architecture distribuée proposée pour TDABC

### 4.3.2. Protocole de communication

L'idée consiste à classer la population initiale selon des sous-populations plus petites et à exécuter simultanément les tâches correspondantes sur différents nœuds. Par exemple, une population de 200 solutions générées par le Nœud.TD est répartie sur 10 Nœud.ABC. Chaque Nœud.ABC va traiter une sous-population de 20 individus.

Le même protocole utilisé dans DVAG est utilisé dans TDABC. La figure 4.13 schématise un scénario de communication entre un Nœud.TD et un Nœud.ABC.

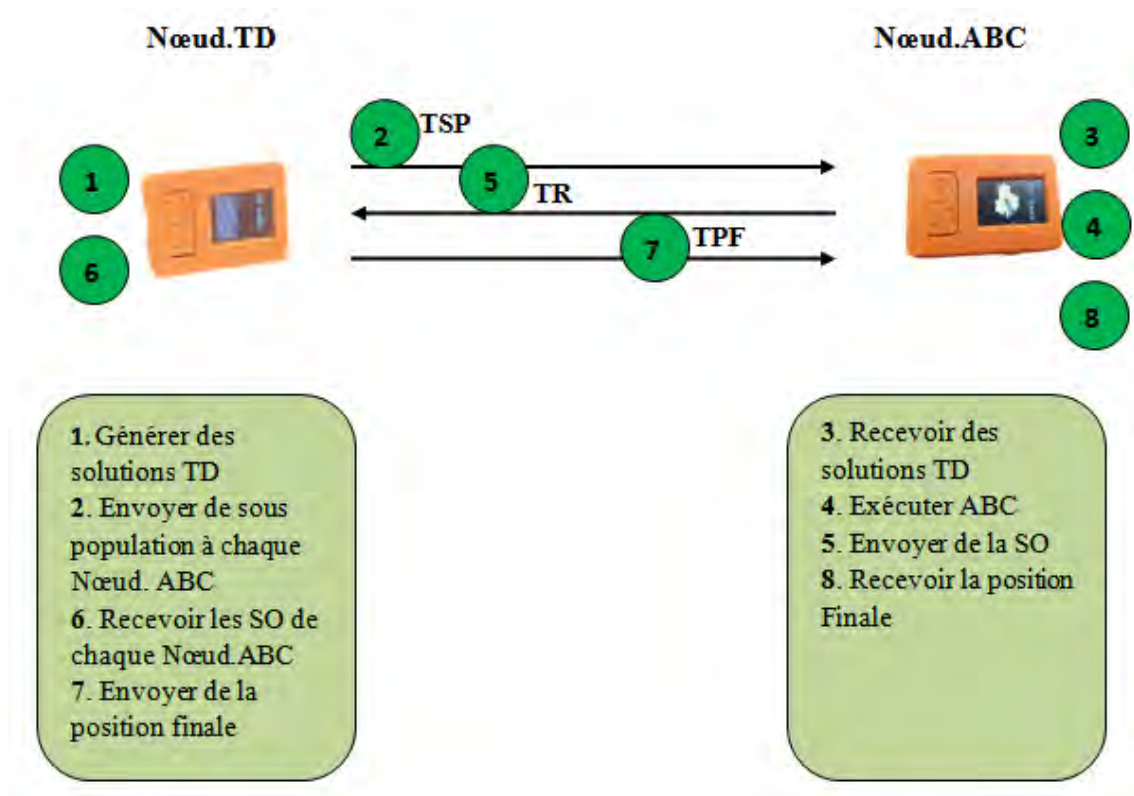


Figure 4. 13 Communication entre un Nœud.TD et un Nœud.ABC

Le protocole de communication est à la base de 4 types des trames, chaque trame a une charge utile allant jusqu'à 250 octets :

- TSP : une Trame-Sous-Population qui contient M solutions de Delaunay, ce type de trame sera envoyé du Nœud.TD vers tous les Nœud.ABC. La figure 4.14 présente la forme générale d'une sous-population de solution de Delaunay.
- TR : une Trame-Résultat qui contient la meilleure solution trouvée pour chaque Nœud. Après l'exécution de son algorithme, chaque Nœud.ABC renvoie une trame TR au Nœud.TD.

- TPF : une Trame-Position-Finale contenant la position finale de chaque nœud, le Nœud.TD rassemble toutes les trames TR renvoyées par chaque Nœud.ABC et prend ainsi la décision de la meilleure solution qui présente la couverture maximale trouvée.
- ACK : ou Accusé de réception, transmis dans les deux sens, et utilisée pour informer la couche protocolaire du succès ou de l'échec de la transmission de chaque type de trame.

Dans l'annexe 1, un diagramme de séquence de notre protocole de communication appliqué sur un exemple de trois nœuds pour l'hybridation TDABC, chaque nœud a un rôle de transmission et de réception (T/R) des trames des données.

La quantité d'information la plus grande se situe au niveau de la trame TSP, la trame TSP sera donc là aussi subdivisée en des sous-trames, en suivant les principes de segmentation de TCP.

	Nœud n°1	Nœud n°2	Nœud n°3	...	Nœud n°M
STD1	$(x_{11}, y_{11})$	$(x_{12}, y_{12})$	$(x_{13}, y_{13})$	...	$(x_{1M}, y_{1M})$
STD2	$(x_{21}, y_{11})$	$(x_{22}, y_{22})$	$(x_{23}, y_{23})$	...	$(x_{2M}, y_{2M})$
⋮					
STDn	$(x_{n1}, y_{n1})$	$(x_{n2}, y_{n2})$	$(x_{n3}, y_{n3})$	...	$(x_{nM}, y_{nM})$

Figure 4. 14 La forme générale d'une sous-population de solutions TD

### 4.3.3. Conception de l'algorithme de positionnement distribué

#### 4.3.3.1. Scénario de déploiement

Le fonctionnement de notre approche comme le montre la figure 4.15 commence par la génération des solutions de Delaunay dans le Nœud.TD, puis ce même nœud subdivise la population générée en des sous-populations. Chaque sous-population sera affectée à un Nœud.ABC. Le processus développe indépendamment sa population jusqu'à ce qu'il décide de rassembler ses meilleurs individus qui seront candidats à l'étape de décision.



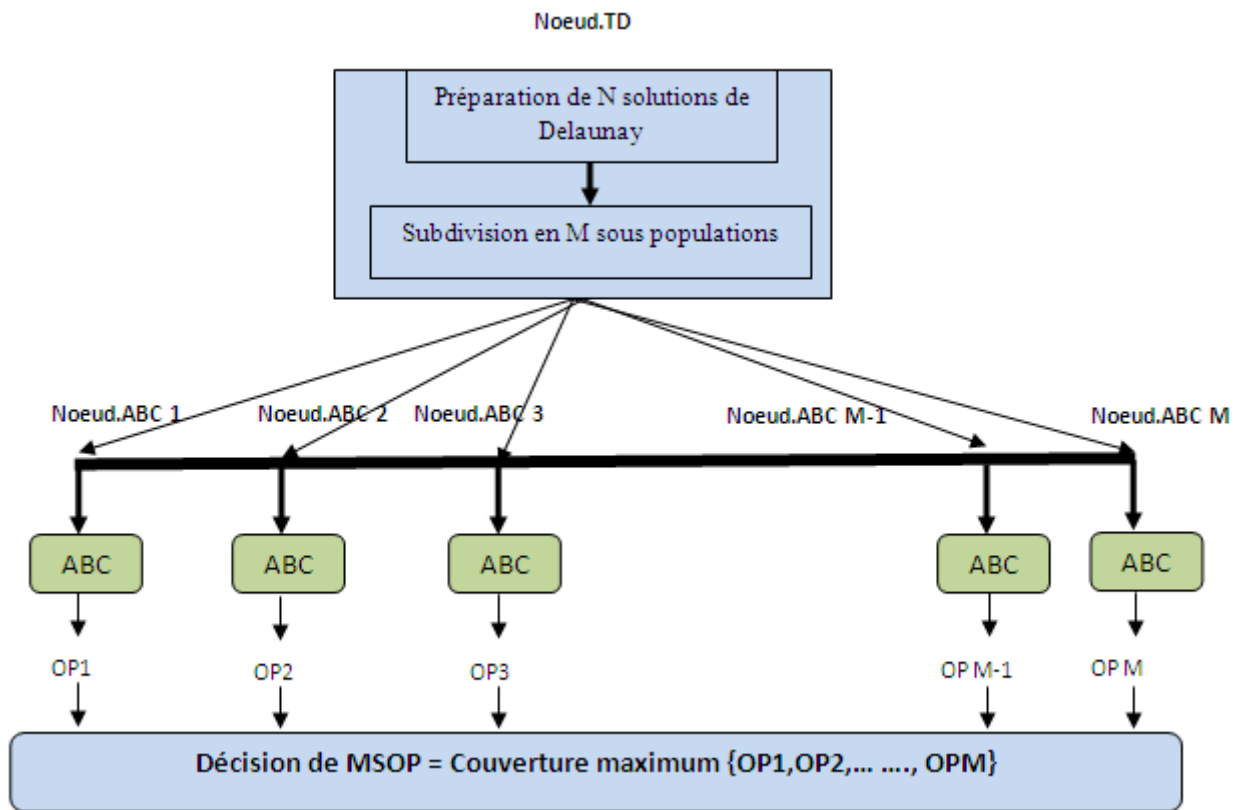


Figure 4. 15 Fonctionnement d'approche proposée

#### 4.3.3.2. Algorithme d'optimisation hybride

L'algorithme 4-2 illustre le pseudo-code de notre algorithme appliqué pour développer l'approche distribuée de TDABC. Cet algorithme commence par l'implémentation de la Triangulation de Delaunay pour générer une population initiale de solutions de Delaunay (Popinitial) et la subdiviser en des sous-populations (lignes 1 et 2). Ensuite, nous passons à l'algorithme ABC exécuté en parallèle dans tous les Nœud.ABC jusqu'à ce qu'une couverture maximale souhaitée soit obtenue (l'étape4 de l'algorithme). Les entrées de l'algorithme TDABC distribuée sont :

- le nombre de nœuds à déployer M,
- le nombre de solutions à générer par le Nœud.TD, N-solutions présentent la population initiale,
- les dimensions de la région d'intérêt (RoI),
- la taille de la sous-population pop\_size,
- la condition d'arrêt (soit par un taux de couverture cherché soit par un nombre d'itération définis dès le début),
- le nombre d'essai I maximal pour abandonner une solution

```
0) Définir la fonction d'évaluation f (Taux de Couverture maximale cherché)
1) Générer N solution de Delaunay Popinitial {Si= S1, S2, S3...SN} dans un RoI
2) Subdiviser la population en pop_size sous population
3) Envoyer une sous population à chaque Nœud.ABC
4) Faire en parallèle sur chaque Nœud.ABC

    4.1) Calculer la couverture de chaque solution Si
    4.2) Evaluer la fonction fitness fi de chaque solution
    4.3) Cycle = 1 ;
    4.4) Répéter
        Pour chaque abeille employée
            {
                Produire une nouvelle solution
                Calculer la valeur fi
            }
        Appliquer le processus de sélection de glouton
        }
        Calcul de probabilité Pi de chaque solution Si
        Pour chaque abeille spectatrice
            {
                Sélection d'une solution Si selon la probabilité Pi
                Produire une nouvelle solution
            }
        Calculer la valeur fi
        Appliquer le processus de sélection de glouton
        }
        S'il existe une solution abandonnée (selon l)

        Alors remplacer-la par une nouvelle solution produite aléatoirement
        Mémoriser la meilleure solution
        Cycle ++ ;
    4.5) Jusqu'à cycle = Nombre de cycle défini ou jusqu'à une couverture maximale
    trouvée
    4.6) Retourner la Meilleure solution SOPi au Nœud.TD.
5) Fin du parallélisme,
6) Meilleure Solution de {SOP1, SOP2.....SOPM}
7) Affectation de position Finale pour chaque nœud
```

#### Algorithme 4- 2 Pseudo-code de TDABC distribuée

Le fonctionnement de l'algorithme de Triangulation de Delaunay et l'algorithme d'optimisation ABC ont déjà été présentés à la troisième section de chapitre 3.

#### 4.3.4. Résultats et discussions

Un véritable prototypage proposé pour l'évaluation de notre approche distribuée sur l'algorithme TDABC, est expérimenté dans un environnement réel avec des expérimentations réalisées sur un banc d'essai contenant jusqu'à 10 nœuds. Nous commençons tout d'abord par les simulations.

#### 4.3.4.1. Simulation

Afin de réaliser les simulations, nous avons utilisé de nouveau OMNET++ : le même simulateur utilisé pour élaborer l'approche distribuée de l'algorithme DVAG. Le choix des valeurs des paramètres repose sur des tests empiriques.

Le processus de distribution s'effectue dans les deux types de nœuds comme suit :

- Pour le nœud Nœud.TD : nous appliquons la Triangulation de Delaunay sur un ROI de 80\*80 mètres, le TD génère une population de 100 solutions (individus) qui présentent les positions de chaque objet IoT dans un plan 2D. Ensuite cette population se subdivise en des sous-populations sur 20 nœuds. Le tableau 4.7 montre les paramètres de simulation utilisés pour le Nœud.TD.

**Tableau 4. 7 Paramètres de simulation de TDABC pour Nœud.TD**

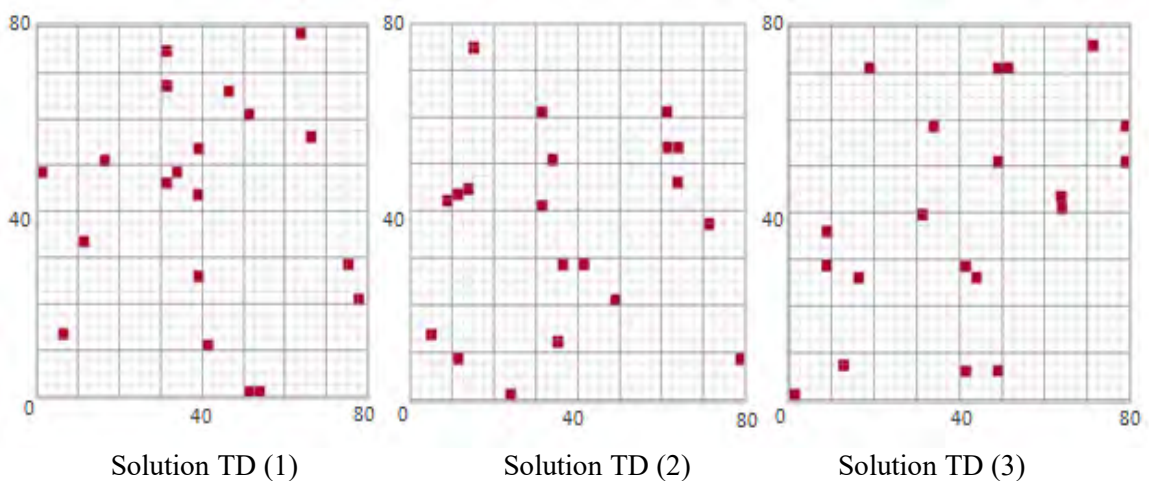
Paramètres	
Taille de la population: Nombre des solutions Delaunay à générer	100..1000
Taille ROI	80*80
Nombre des objets à déployer (déploiement initiale des nœuds)	20
Taille de chaque sous population	10..100
RSSI	100 (initial)
FER	0.01(initial)
Portée de détection intérieure	10 m
Fréquence	2.4 GHz
Moyenne du nombre d'exécution	30
Antenne	isotropique
Température de fonctionnement du nœud	0° C à 60° C

- Pour le Nœud.ABC : après la réception des solutions de Delaunay, nous exécutons l'algorithme d'optimisation ABC sur chacun des Nœud.ABC en parallèle sur le même ROI de 80\*80. Le ABC permet d'optimiser les solutions trouvées par le TD (déploiement initial) en remplaçant les positions des nœuds afin de trouver le meilleur emplacement dans le RoI et avoir par la suite le degré de couverture maximal cherché. Le tableau 4.8 montre les paramètres de simulation utilisés pour chaque Nœud.ABC.

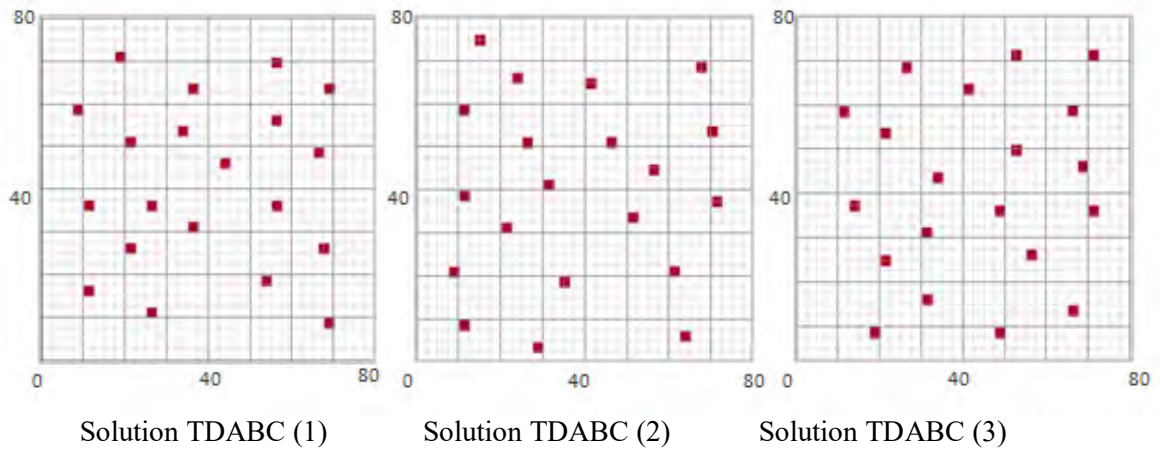
**Tableau 4. 8 Paramètres de simulation de TDABC pour les Nœud.ABC**

Paramètres	
Taille de la population reçue	10...100
Nombre d'itération	[100..1500]
Nombre des objets à déployer (nœuds)	20
Dimension ROI	80*80
Limite l " des abeilles éclaireuses "	30
Portée de détection d'un nœud	10
Condition d'arrêt (taux de couverture)	94%
RSSI	100 (initial)
FER	0.01(initial)
Fréquence	2.4 GHz
Moyenne du nombre d'exécution	30
Antenne	isotropique
Température de fonctionnement du nœud	0° C à 60° C

La figure 4.16 et la figure 4.17 montrent clairement la différence, en termes de taux de couverture, entre la couverture aléatoire initiale générée uniquement par le diagramme de Delaunay et le degré de couverture fourni par le TDABC proposé.

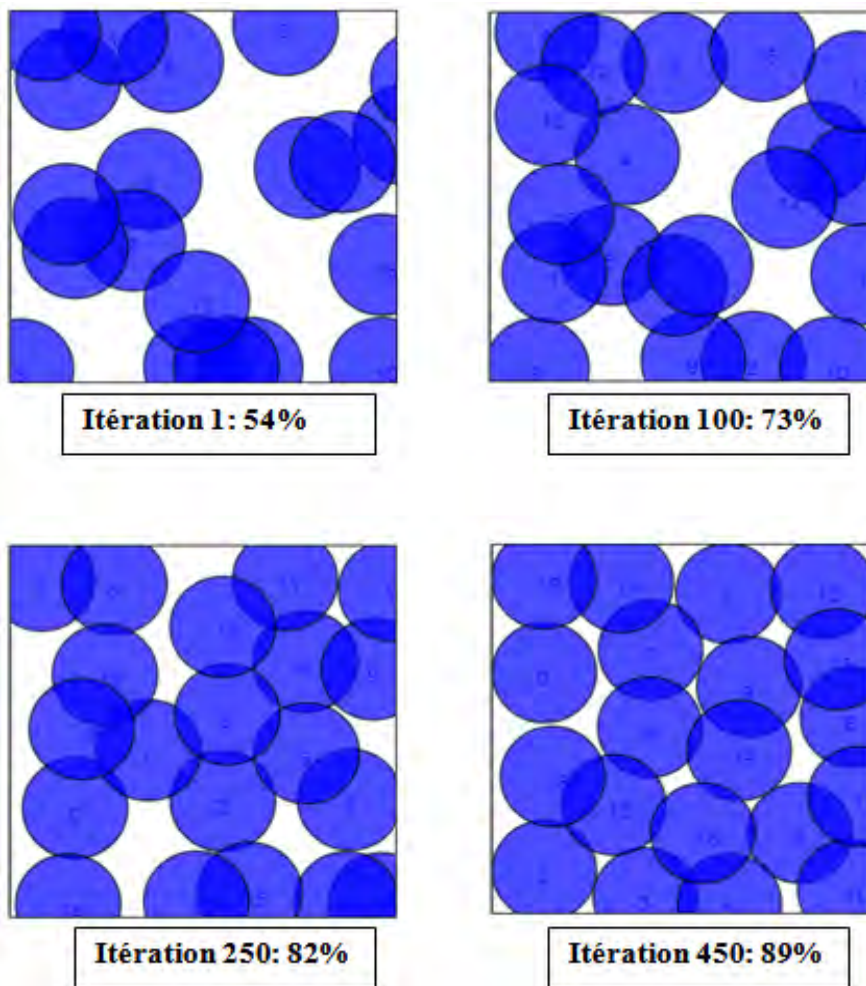


**Figure 4. 16 Déploiement de 20 nœuds avec TD seulement (exemple de 3 solutions)**



**Figure 4. 17 Déploiement de 20 nœuds après l'exécution de TDABC (exemple de 3 solutions)**

La figure 4.18 montre l'augmentation de taux de couverture après chaque itération de TDABC.



**Figure 4. 18 Déploiement de 20 nœuds dans un champ 80\*80 par TDABC**

La figure 4.19 présente le degré de couverture pour six nœuds utilisés dans une approche distribuée de TDABC, chaque nœud atteint sa couverture désirée (94%) avec un nombre d'itération différent d'un autre nœud, par exemple le nœud 2 a trouvé sa couverture dans l'itération 650 tandis que le nœud 6 a trouvé sa couverture à l'itération 850.

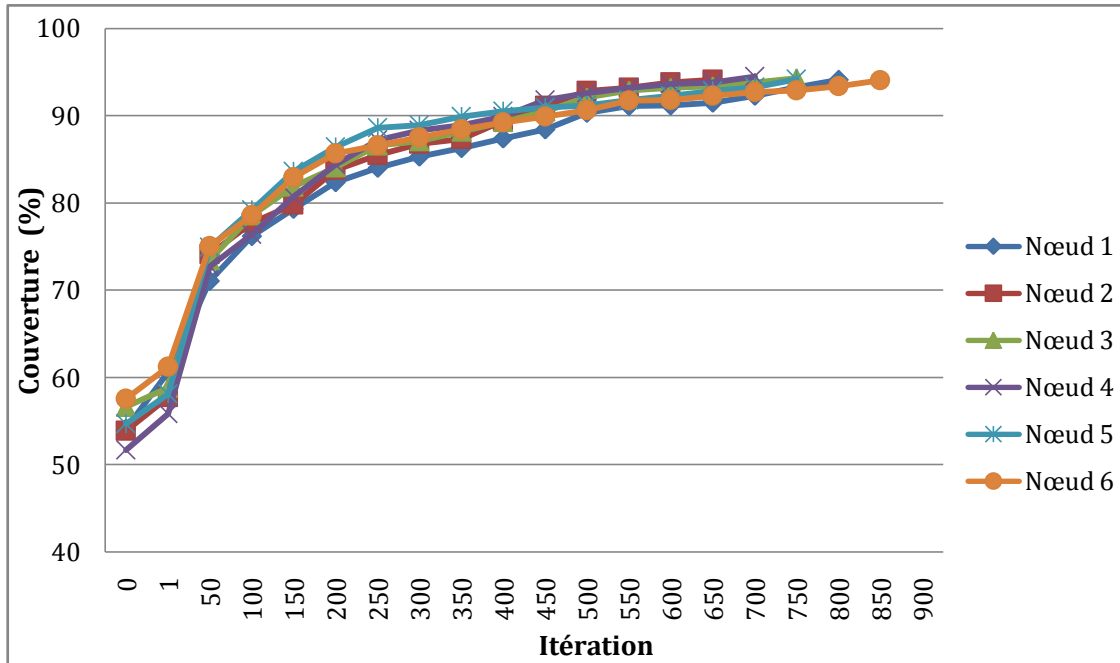


Figure 4. 19 Degré de couverture dans l'approche distribuée pour six nœuds

#### 4.3.4.2. Expérimentations

Nous utilisons le même type de nœud M5StickC dans nos expérimentations. Un véritable banc d'essai mettant en œuvre plusieurs nœuds réels, est conçu et organisé dans un réseau sans fil.

##### 4.3.4.2.1. Premier cadre d'expérimentation

Tableau 4. 9 Les paramètres de premier cadre d'expérimentation

Paramètres	
Taille de la population	100
Nombre d'itération	[600..900]
Nombre des objets à déployer (nœuds)	06
Dimension de ROI	25*12*3 (2 étages)
Limite l " des abeilles éclaireuses "	25
Condition d'arrêt (taux de couverture)	94%

Seuil de résultats reçus (en %)	80%
RSSI	100 (initial)
FER	0.01(initial)
Portée de détection intérieure	10 m
Fréquence	2.4 GHz
Moyenne du nombre d'exécution	30
Antenne	isotropique
Température de fonctionnement du nœud	0° C to 60° C

Dans ce premier temps, nous avons utilisé le même cadre d'expérimentation de DVAG centralisée. Le cadre des expérimentations est défini comme suit : 6 nœuds M5StickC sont déployés dans un appartement composé d'un rez-de-chaussée et de 2 étages chacun a une surface de 25\* 17 mètres comme le montre la figure 4.20.

- Les taux de RSSI et FER sont calculés. La puissance de transmission du WiFi est toujours de 100 mW.
- En raison de la nature stochastique des algorithmes d'optimisation, le processus d'optimisation est exécuté 30 fois et une valeur moyenne est calculée pour toutes les valeurs des figures.

Nous ajoutons le paramètre "Seuil de résultat reçu", car la plupart des nœuds ABC trouvent leurs solutions optimales dans un délai similaire, alors que d'autres prennent plus de temps pour la récupération des données pour prendre la décision afin d'avoir la meilleure solution.

**Tableau 4. 10 Emplacements des nœuds dans le premier cadre d'expérimentation**

Nœud	Position
Nœud n°1	(9.20; 5.69; 2.25)
Nœud n°2	(08.37; 12.50; 1.50)
Nœud n°3	(10.50;19.80; 2.20)
Nœud n°4	(10.20; 08.30; 1.75)
Nœud n°5	(9.50; 18.50; 1.70)
Nœud n°6	(10.50; 13.20; 00.00)

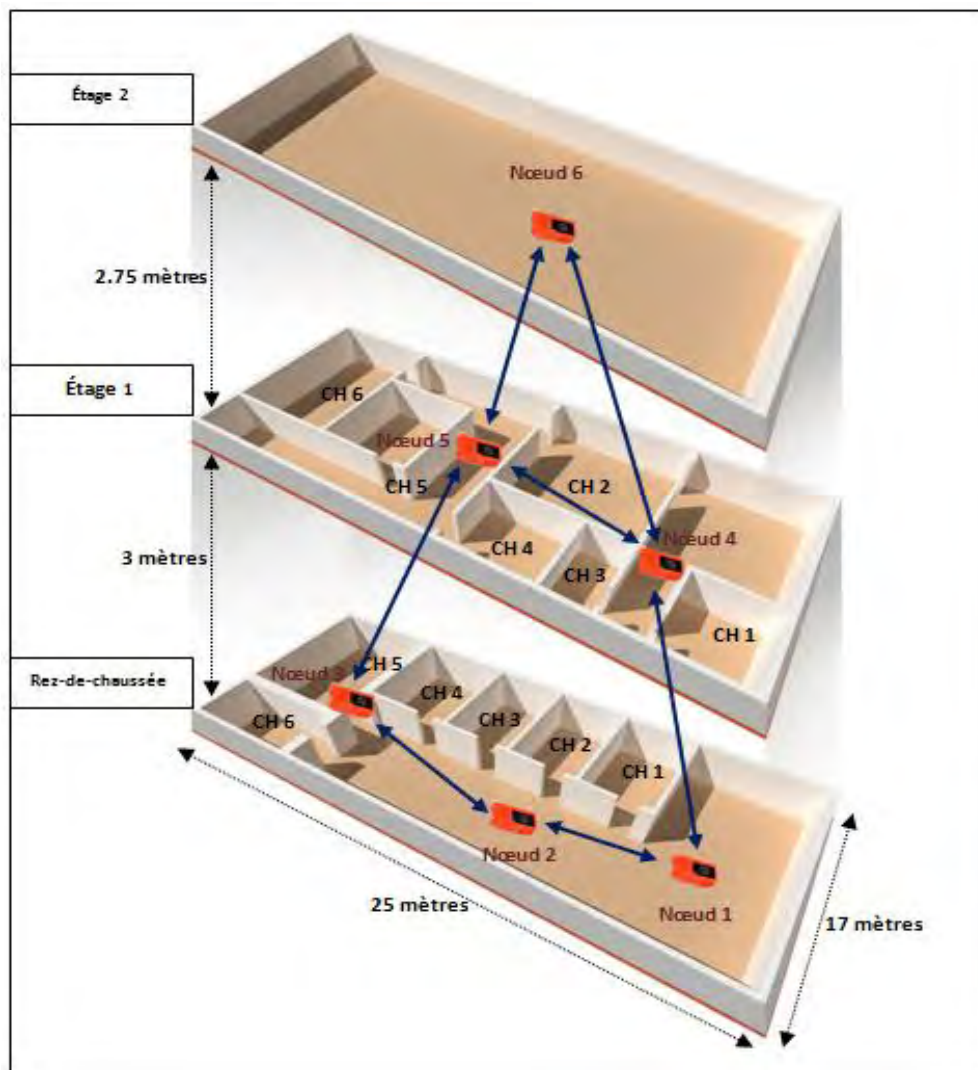


Figure 4. 20 Environnement n°1 d'expérimentation

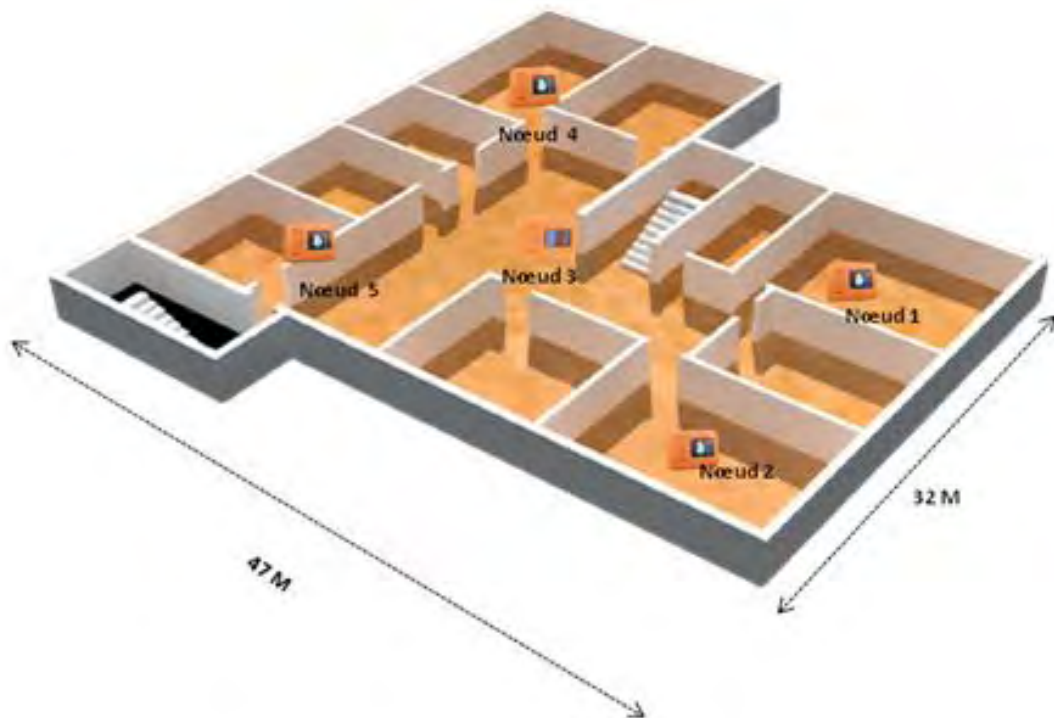
#### 4.3.4.2.2. Deuxième cadre d'expérimentation

Le deuxième cadre des expérimentations utilisé est le rez-de-chaussée de l'Institut Supérieur des Arts et Métiers de Gafsa, la topologie est définie comme suit : 05 nœuds M5StickC sont déployés dans une surface de 47 x 32 mètres comme le montre la figure 4.21.



**Tableau 4. 11 Les paramètres du deuxième cadre d'expérimentation**

<b>Paramètres</b>	
Taille de la population	100
Nombre d'itération	[600..900]
Nombre des objets à déployer (nœuds)	05
Dimension de ROI	47*32*3
Limite l " des abeilles éclairées "	25
Condition d'arrêt (taux de couverture)	94%
Seuil de résultats reçus (en %)	80%
RSSI	100 (initial)
FER	0.01(initial)
Portée de détection intérieure	10 m
Fréquence	2.4 GHz
Moyenne du nombre d'exécution	30
Antenne	isotropique
Température de fonctionnement du nœud	0° C à 60° C



**Figure 4. 21 Environnement n°2 d'expérimentation de TDABC**

**Tableau 4. 12 Emplacements des nœuds dans le deuxième lien d'expérimentation**

Nœud	Position
Nœud n°1	(25.30; 10.70; 1.50)
Nœud n°2	(10.37; 10.50; 2.50)
Nœud n°3	(17.50; 31.75; 2.30)
Nœud n°4	(27.20; 36.50; 1.75)
Nœud n°5	(11.50; 35.50; 1.70)

**Evaluation de l'algorithme TDABC selon les métriques du réseau (RSSI, durée de vie et nombre de voisins)**

Les tableaux 4.13, 4.14 et 4.15 représentent, respectivement, les valeurs moyennes du RSSI, de la durée de vie et du nombre de voisins des nœuds utilisés

**Tableau 4. 13 Valeurs moyennes du RSSI pour 6 nœuds**

Noeud	TD	ABC	TDABC
1	122	132	133
2	134	143	139
3	128	124	136
4	141	132	137
5	131	128	144
6	126	134	128
<b>Moyenne des valeurs</b>	130.3333333	132.1666667	136.1666667

Le tableau 4.13 montre des valeurs élevées de RSSI issues des nœuds positionnés selon l'algorithme TDABC.

**Tableau 4. 14 Valeurs moyennes de durée de vie (en secondes) pour 6 nœuds**

Noeud	TD	ABC	TDABC
1	1187	1387	1826
2	1229	1384	1627
3	1327	1256	1734
4	1252	1285	1725
5	1233	1391	1619
6	1324	1342	1862
<b>Moyenne des valeurs</b>	1258.67	1340.83	1732.17

Le tableau 4.14 illustre les durées de vie élevées des nœuds positionnés selon l'algorithme TDABC. Il indique une meilleure capacité de couverture (déploiement) des nœuds à l'aide de cet algorithme.

**Tableau 4. 15 Valeurs moyennes du nombre de voisins pour 6 nœuds**

Node	TD	ABC	TDABC
1	4.32	5.42	5.42
2	4.56	4.48	4.29
3	5.33	3.16	5.27
4	3.44	4.12	4.91
5	4.51	4.69	4.56
6	4.61	4.94	4.82

Moyenne des valeurs	4.46167	4.46833	4.87833
---------------------	---------	---------	---------

Le tableau 4.15 indique des nombres élevés de voisins des nœuds déployés selon l'hybridation TDABC. La supériorité de ce nombre de voisins reflète la meilleure qualité de connectivité des nœuds utilisant dans cette hybridation.

#### 4.4. D-DVAG vs D-TDABC

Dans cette partie nous allons comparer notre approche distribuée pour les deux hybridations DVAG et TDABC en termes de taux de couverture, coût temporel, valeurs RSSI et la durée de vie.

##### 4.4.1. Taux de couverture

Nous mesurons le pourcentage de couverture de surface dans les 2 hybridations, avec une condition d'arrêt d'un taux de couverture égale à 94%. Dans les deux hybridations nous avons cumulé la moyenne des résultats trouvés qui atteint cette condition d'arrêt. DVAG atteint une couverture de 94.36% dans l'itération 755 alors que TDABC atteint le 94.10 % à l'itération 701 comme le montre la figure 4.22.

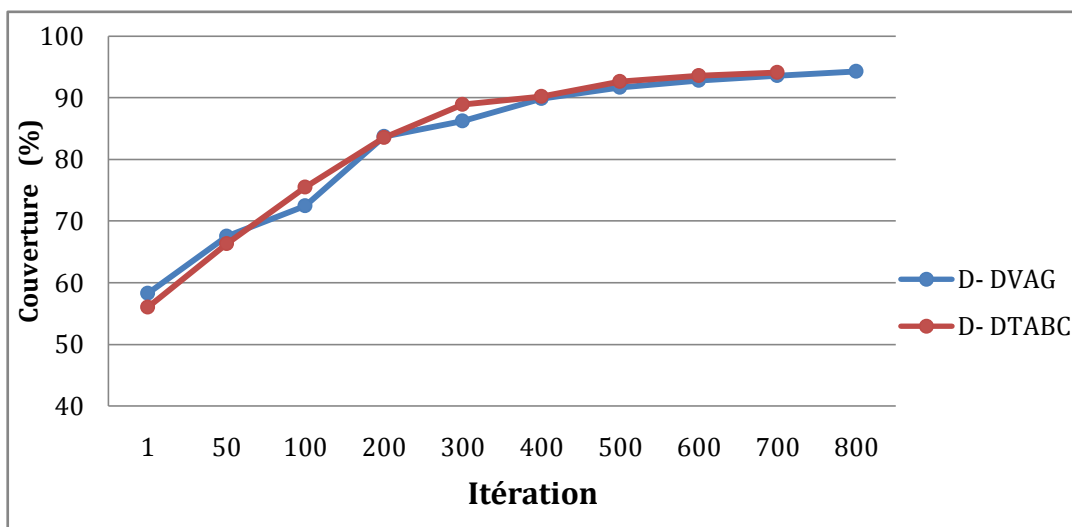
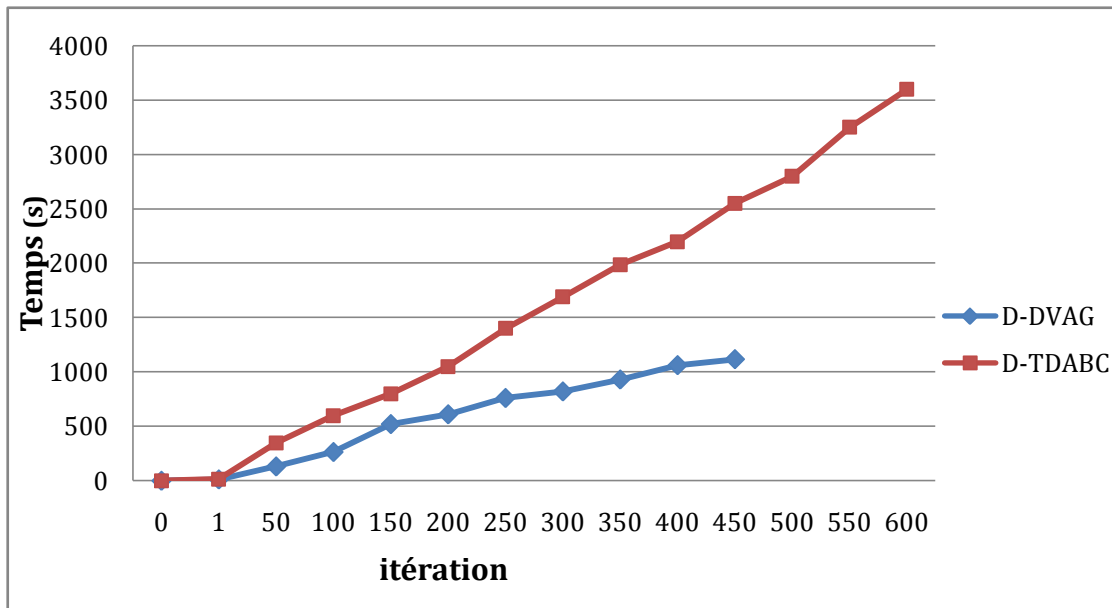


Figure 4. 22 Comparaison du degré de couverture entre DVAG et TDABC dans une approche distribuée

##### 4.4.2. Coût de temps de calcul

L'hybridation distribuée TDABC a un coût de temps de calcul très élevé par rapport à l'hybridation distribuée de DVAG, comme le montre la figure 4.23. Cela s'explique par le coût de calcul de chaque itération dans l'algorithme ABC, il y a un parcours de toutes les solutions dans les trois phases de cet algorithme avec le calcul de la fonction fitness pour chaque individu.



**Figure 4. 23 Comparaison du coût de temps de calcul entre DVAG et TDABC dans une approche distribuée**

#### 4.4.3. RSSI, durées de vie et nombres des voisins

Les tableaux 4.16, 4.17 et 4.18 représentent les valeurs moyennes du RSSI, de la durée de vie et du nombre de voisins des nœuds utilisés.

**Tableau 4. 16 Valeurs moyennes du RSSI pour 6 nœuds**

Noeud	DV	TD	AG	ABC	DVAG	TDABC
1	132.39	122	131.83	132	137.28	133
2	135.28	134	128.92	143	136.64	139
3	131.23	128	125.17	124	131.55	136
4	114.72	141	116.28	132	115.21	137
5	89.29	131	92.88	128	91.53	144
6	129.82	126	127.76	134	129.16	128
<b>Moyenn e</b>	122.12166 67	130.33333 33	120.47333 33	132.16666 67	123.56166 67	136.16666 67

Le tableau 4.16 montre des valeurs élevées de RSSI issues des nœuds positionnés selon l'algorithme DVAG. Comme déjà dit, la supériorité de ces valeurs RSSI indique une meilleure capacité de localisation des nœuds à l'aide de cet algorithme.

**Tableau 4. 17 Valeurs moyennes de durée de vie (en secondes) pour 6 nœuds**

Noeud	DV	TD	AG	ABC	DVAG	TDABC
1	1298	1187	1367	1387	1693	1826

2	987	1229	927	1384	996	1627
3	1029	1327	1035	1256	1083	1734
4	703	1252	749	1285	753	1725
5	870	1233	883	1391	892	1619
6	943	1324	932	1342	921	1862
<b>Moyenne</b>	971.66	1258.67	982.16	1340.83	1056.33	1732.17

Le tableau 4.17 illustre les durées de vie élevées des nœuds positionnés selon l'algorithme DVAG. Cela indique une meilleure capacité de couverture (déploiement) des nœuds à l'aide de cet algorithme.

**Tableau 4. 18 Valeurs moyennes du nombre de voisins pour 6 nœuds**

Node	DV	TD	AG	ABC	DVAG	TDABC
1	4.23	4.32	4.65	5.42	4.98	5.42
2	3.22	4.56	3.41	4.48	3.56	4.29
3	2.68	5.33	2.91	3.16	3.02	5.27
4	3.56	3.44	3.86	4.12	3.94	4.91
5	4.29	4.51	4.44	4.69	4.62	4.56
6	3.12	4.61	3.42	4.94	3.68	4.82
<b>Moyenne</b>	3.5166	4.4616	3.7816	4.4683	3.9666	4.8783

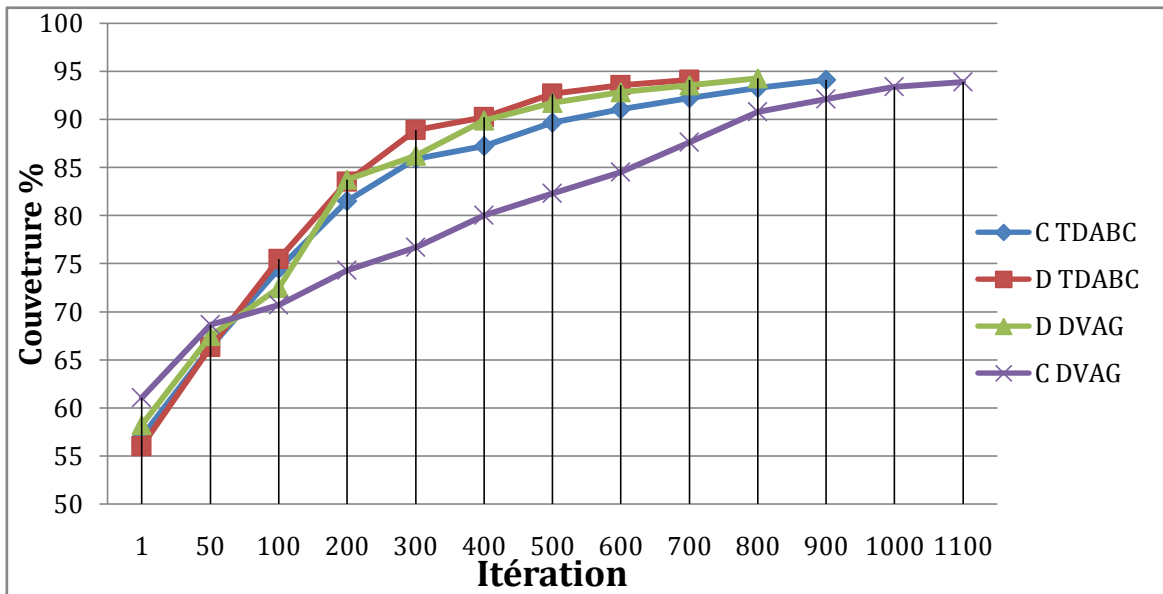
Le tableau 4.18 indique des nombres élevés de voisins des nœuds déployés selon l'hybridation DVAG. De nouveau, la supériorité de ce nombre de voisins reflète la meilleure qualité de connectivité des nœuds utilisant dans cette hybridation

#### **4.5. Approche centralisée vs l'approche distribuée**

Dans cette section, nous comparons les résultats trouvés dans notre approche distribuée par rapport à une approche centralisée (chapitre 3) en termes de pourcentage de couverture trouvée et de temps d'exécution. Une deuxième comparaison concerne la consommation d'énergie dans les deux approches. Ces résultats sont obtenus avec une moyenne d'exécution de 40 itérations.

##### **4.5.1. Taux de couverture**

L'objectif de ces évaluations est de comparer notre approche distribuée avec l'approche centralisée.



**Figure 4. 24 Comparaison du degré de couverture entre l'approche distribuée et l'approche centralisée**

Nous mesurons le pourcentage de couverture de surface dans les 2 approches, avec une condition d'arrêt de 94% de couverture. Dans l'approche distribuée, nous cumulons la moyenne des résultats reçus de différents nœuds permettant d'atteindre cette couverture. Nous pouvons clairement voir qu'une approche distribuée pour DVAG atteint l'objectif dans les 800ième générations tandis que l'approche centralisée continue à fonctionner pendant 310 itérations supplémentaires comme le montre figure 4.24. De même pour une approche distribuée de TDABC atteint l'objectif dans les 709ième générations tandis que l'approche centralisée continue à fonctionner pendant 200 itérations supplémentaires.

#### **4.5.2. Coût du temps de calcul**

Bien que dans une approche distribuée, il y ait un coût de temps de calcul, à savoir le temps de distribution des sous-populations pour chaque nœud, le temps d'exécution et le temps de renvoi de solution trouvée, le graphique de la figure 4. 25 montre clairement que l'approche distribuée a terminé son fonctionnement dans un délai inférieur face à une approche centralisée dans les deux hybridations D-DVAG et D-TDABC. Ceci qui s'explique par le coût de calcul de la fonction fitness en rapport à chaque individu dans une grande population utilisée dans l'approche centralisée.

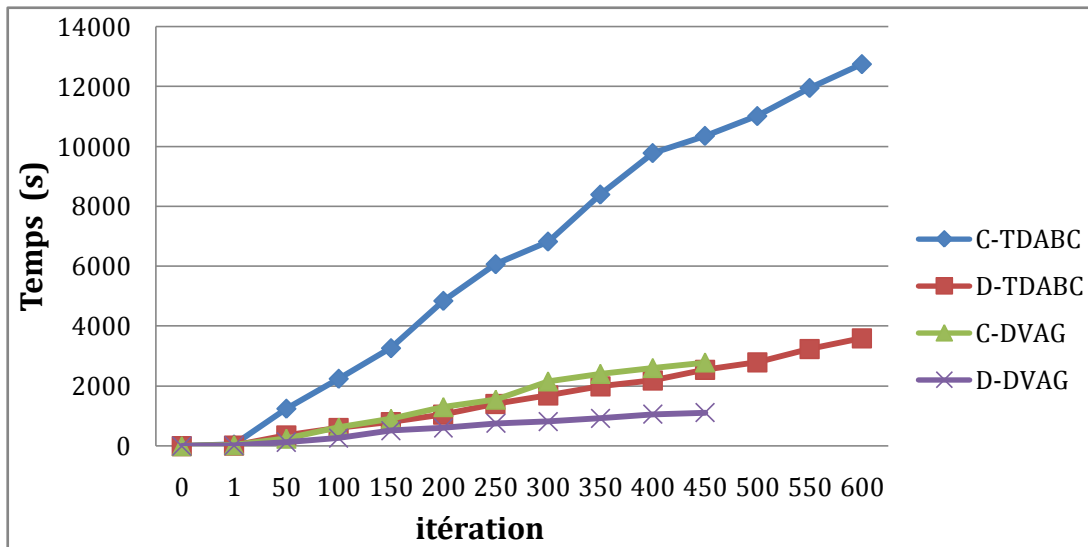


Figure 4. 25 Comparaison du coût de calcul de temps entre l'approche distribuée et l'approche centralisée

### 4.5.3. Comparaison en termes de consommation d'énergie

La consommation d'énergie est un paramètre très utilisé pour comparer les deux approches. Pour cela nous avons placé tous les nœuds en mode plein charge (100%) dans le même instant. Pour l'approche distribuée nous avons fait la moyenne de consommation de tous les nœuds utilisées dans cette approche. Le tableau 4.19 montre une comparaison entre les deux approches, un nœud M5StickC dans C-DVAG a consommé d'environ 54% d'énergie de sa batterie jusqu'à la fin de son fonctionnement et de 28% pour l'approche distribuée (D-DVABC).

Tableau 4. 19 Niveau de charge de batterie

Temps (S)	0	500	1000	1500	2000	2500	3000	6000	8000	20000	25000
<b>C-DVAG</b>	100%	96%	86%	71%	65%	57%	<b>46%</b>	-----	-----	-----	-----
<b>D-DVAG</b>	100%	96%	83%	77%	<b>72%</b>	-----	----	-----	-----	-----	-----
<b>C-TDABC</b>	100%	81%	69%	58%	45%	39%	31%	10%	1%	0%	<b>0%</b>
<b>D-TDABC</b>	100%	83%	71%	64%	55%	51%	46%	29%	<b>8%</b>	-----	-----



Un nœud M5StickC dans C-TDABC a consommé toute l'énergie de sa batterie avant de terminer son fonctionnement, tandis que pour l'approche distribuée (D-TDABC) a consommé d'environ 92% jusqu'à la fin de son fonctionnement.

Nous constatons que dans l'approche distribuée pour les deux hybridations D-DVAG et D-TDABC consomme moins d'énergie qu'une approche centralisée (C-DVAG et C-TDABC).

Pour ces calculs et interprétations, nous supposons que toutes les batteries sont équivalentes, ce qui est globalement le cas dans une première approche. Certaines batteries étant plus vieilles que d'autres, il pourrait être intéressant de mener une mesure plus fine, avec des équipements électroniques dédiés, hors du champ de compétences de cette thèse.

#### **4.6. Conclusion**

Dans ce chapitre nous parvenons à décrire et présenter notre approche distribuée où l'exécution se fait dans plusieurs nœuds en parallèle. Nous avons utilisé cette approche pour résoudre le problème de déploiement optimisé dans les réseaux sans fils de collecte des données pour nos deux hybridations DVAG et TDABC.

Premièrement, nous avons présenté pour chaque hybridation la stratégie de distribution, le protocole de communication et le scénario de déploiement ainsi que les résultats obtenus. Deuxièmement nous avons comparé les deux hybridations dans la même approche distribuée D-DVAG et D-TDABC en termes de taux de couverture, coût de temps de calcul et les métriques RSSI, durée de vie et nombre de voisinage. Finalement nous avons comparé les résultats obtenus dans l'approche centralisée (chapitre 3) par les résultats obtenus par l'approche distribuée.

Pour une approche distribuée les hybridations DVAG et TDABC atteints une couverture de 94% dans un nombre d'itération très proche. L'algorithme distribué de DVAG (D-DVAG) proposé surpasse celui de l'algorithme distribué de TDABC (D-TDABC) en termes de vitesse de calcul.

## Conclusion générale et perspectives

### Apports de la thèse

Notre thèse de recherche a étudié le problème de déploiement 3D des nœuds dans les réseaux sans fil de collecte de données en développant deux approches d'optimisation. Ces deux approches sont basées sur l'hybridation entre les méthodes géométriques de déploiement et les algorithmes d'optimisations pour l'amélioration de taux de couverture avec un minimum des nœuds.

Les contributions principales et les conclusions fondamentales de nos travaux de recherche pourraient être résumées comme suit :

- une étude bibliographique approfondie des méthodes d'optimisation du déploiement dans les réseaux sans fil de collecte des données, notamment des algorithmes prenant en compte le déploiement 2D et 3D à l'intérieur. Cette étude couvre les méthodes classiques de déploiement telles que les techniques de Grille, les techniques basées sur la géométrie computationnelle et les techniques de force virtuelles. Cette étude couvre aussi les méta-heuristiques avec des nombreux algorithmes d'optimisation stochastique qui ont été conçus pour résoudre différents problèmes d'optimisation. L'étude traite également du cas mono-objectif et multi-objectifs, le cas statique et dynamique du déploiement,
- une première hybridation nommée DVAG combine une méthode géométrique de déploiement présentée par le Diagramme de Voronoi DV avec un algorithme d'optimisation présenté par l'algorithme génétique AG. Le DV a pour but de diviser le champ d'intérêt en cellules et d'engendrer des solutions initiales sur les positions de chaque nœud, puis un algorithme génétique AG est utilisé pour déterminer les meilleurs emplacements, à partir de résultats obtenus par le DV, afin de maximiser la couverture,
- une deuxième hybridation nommée TDABC combine une méthode géométrique de déploiement présentée par le diagramme de Triangulation de Delaunay TD avec un algorithme d'optimisation présenté par l'algorithme de colonie des abeilles artificielles ABC. Le TD a pour but de générer des solutions qui présentent un déploiement initial des objets IoT, puis un algorithme ABC sera exécuté pour améliorer le taux de couverture de ce déploiement des nœuds,
- une approche distribuée pour la première hybridation DVAG, le diagramme de Voronoi DV présente le point de départ de cette hybridation qui générera une population initiale présentant un déploiement aléatoire d'objets. Cette génération sera exécutée sur un seul nœud et la population sera distribuée sur l'ensemble des autres nœuds formant le réseau de collecte des données. Alors, un algorithme génétique est exécuté dans chacun de ces nœuds dans le but d'optimiser ce déploiement,

- une approche distribuée pour la deuxième hybridation TDABC commence par la génération de la population initiale à travers la Triangulation de Delaunay TD. Cette génération sera exécutée dans un seul nœud et la population générée sera distribuée sur l'ensemble des autres nœuds formant le réseau de collecte des données RCD. Ceci qui fait que l'algorithme ABC est exécuté en parallèle dans chacun de ces nœuds,
- nous avons alors travaillé à implémentation d'un protocole de communication entre les nœuds avec une stratégie de distribution pour les deux hybridations. Ce protocole de communication a pour but de gérer la communication entre les différents nœuds utilisés pour le déploiement,
- les approches proposées ont été évaluées sur un environnement expérimental par prototypage sur un banc d'essai réel à l'aide des nœuds M5StickC équipés d'un processeur ESP32. Ce testbed nous révèle qu'une approche distribuée dépasse celles des algorithmes originaux en termes de degré de couverture, de RSSI, de durée de vie des nœuds et améliore le nombre d'objets voisins, ce qui est positif pour améliorer la qualité des communications, même en cas de défaillances de certaines liaisons radio. Cette approche dépasse aussi une approche centralisée en termes de rapidité de calcul pour la première hybridation DVAG.

Le seul obstacle rencontré est que la taille mémoire d'un nœud M5StickC ne permet pas d'exécuter notre approche lorsque le nombre de nœuds à déployer dépassait 58 nœuds, ce qui est tout de même raisonnable pour un réseau d'intérieur.

Les obstacles dans un champ d'intérêts dans nos travaux ne sont pas pris en compte. En résumé, les indicateurs de performance utilisés pour évaluer nos travaux, les simulations et le prototypage réel mettent en évidence le bon comportement de nos approches proposées sur des problèmes multi-objectifs et montrent l'efficacité pour les deux hybridations.

## Perspectives

Les approches et les hybridations présentées dans notre thèse ont ouvert de nouvelles voies de recherche qui pourraient améliorer le déploiement du réseau de collecte de données au contexte pour effectuer l'optimisation de la couverture. Les perspectives de recherche futures possibles sont les suivantes :

1. définir un cadre pour intégrer les informations spatiales dans les algorithmes de déploiement des réseaux de collecte des données. Nous envisageons un cadre conceptuel définissant la méthode qui permet l'intégration d'informations contextuelles dans le processus d'optimisation d'un RCD. Ces informations devraient inclure des informations spatiales, la topographie de la zone d'étude ainsi que les obstacles naturels et artificiels, ce qui permettrait à l'optimisation des RCD complexe avec des obstacles

environnementaux. Étant donné la rareté des recherches qui ont ciblé l'intégration des caractéristiques géospatiales de l'environnement, le développement de nouvelles approches a un intérêt d'importance pratique lorsqu'elles prennent en compte ces caractéristiques de l'environnement telles que l'hétérogénéité du terrain, le modèle de terrain et les obstacles artificiels et naturels, en particulier pour optimiser la ligne de vue radio entre les objets connectés. Ces informations spatiales doivent être analysées et étudiées au cours du processus d'optimisation,

2. s'intéresser à une meilleure d'optimisation de la consommation de l'énergie présentant une des métriques des plus importantes dans l'évaluation des performances d'un réseau sans fil de collecte de données. En effet, la consommation d'énergie est considérée comme un paramètre fondamental dans le contexte de disponibilité dans les RCD. Ce paramètre pose la problématique de la préservation énergétique particulièrement si l'application doit fonctionner longtemps. Outre cet intérêt, on identifie également l'apport face à l'aspect développement durable et réduction de l'énergie consommée,
3. proposer d'autres schémas d'hybridations entre les techniques classiques de déploiement (telles que les technique de Grille et les techniques de force virtuelle) avec les algorithmes d'optimisation.
4. les hybridations et les approches que nous avons implémentées pour le déploiement à l'intérieur, tentent de répartir les nœuds capteurs sur un champ d'intérêt pour obtenir un degré de couverture souhaité avec l'utilisation du minimum des nœuds. Nous proposons d'évaluer ces approches dans des environnements expérimentaux à grande échelle dans lesquelles le nombre de nœuds dépasse 100 nœuds. La plateforme Locura4IoT intégrée dans FIT-IoT Lab sera utile pour ceci.
5. étant donné que les modèles hybrides ont généralement des complexités élevées [60], nous proposons d'établir une analyse de la complexité temporelle et algorithmique et une étude statistique afin de prouver la scalabilité des modèles proposées,
6. construire des réseaux sans fil de collecte des données avec différentes portées de détection, serait intéressant et pourrait grandement améliorer les résultats de l'optimisation. Un RCD pourrait être composé de plusieurs types de nœuds capteurs avec des spécifications différentes, y compris leur portée de détection et leur modèle de détection (par exemple, un modèle de détection circulaire, ellipsoïdal ou irrégulier), mais aussi des technologies de communication multiples : LoRa, SigFox, Wi-Fi, BLE, ZigBee, UWB... Les apports de notre thèse de recherche portent sur le développement de stratégies de déploiement incluant des réseaux de capteurs homogènes, il sera alors utile de travailler sur les réseaux hétérogènes,

7. le déploiement du réseau de capteurs peut rencontrer des problèmes plus complexes en termes de détection de trous et d'optimisation de la couverture. Dans certaines applications, telles que le contrôle de sécurité, il est nécessaire que chaque point de la région soit couvert par au moins  $k$  ( $k > 1$ ) capteurs. Les solutions proposées dans les chapitres 3 et 4 nous ont amenés à résoudre ce problème de réseaux de capteurs à  $k$ -couverture. L'algorithme proposé peut vérifier le  $k$ -couverture pour la zone, mais le développement des algorithmes pour combler les trous dans une zone de détection donnée reste une question ouverte.

## Publications (dans cette thèse)

### Reuves internationales

(1) – Publication d'un papier revue intitulé : « **Distributed approach for the deployment of wireless connected objects indoors by hybridization between the Voronoi diagram and the genetic algorithm** »

*Abdallah, W., Mnasri, S., & Val, T. (2022). "Distributed approach for the indoor deployment of wireless connected objects by the hybridization of the Voronoi diagram and the Genetic Algorithm", Journal of Engineering Research and Sciences, Volume 1, Issue 2, Page # 10-23, 2022; doi: 10.55708/js0102002*

(2) - Soumission d'un papier intitulé «**Distributed Artificial Bee Colony Algorithm and Delaunay Triangulation for the coverage in IoT networks pandemics**» soumis dans le revue "Computer Networks",

<https://www.journals.elsevier.com/computer-networks>

Auteurs : ABDALLAH Wajih , MNASRI Sami, VAL Thierry,

(3) - Soumission d'un papier intitulé « **BLE devices deployment and localization in indoor common spaces as a Social IoT solution to fight against the COVID-19 and next pandemics** » soumis dans le revue "Computer Networks", ISSN: 1389-1286, (<https://www.scimagojr.com/journalsearch.php?q=26811&tip=sid&clean=0> ),

Auteurs : MNASRI Sami, VAL Thierry, ABDALLAH Wajih

### Conférences internationales

(4) – Publication d'un papier intitulée « **A collaborative talking assistive technology for people with Autism Spectrum Disorders** » In International Conference on Human-Computer Interaction (pp. 3-12), (HCI 2019, 26 - 31 July 2019), Orlando, Florida, United States. pp.3-12,  
Accès: <https://hal.archives-ouvertes.fr/hal-02884692>

*Abdallah, W., Vella, F., Vigouroux, N., Bossche, A. V. D., & Val, T. (2019, July). A collaborative talking assistive technology for people with Autism Spectrum Disorders. In International Conference on Human-Computer Interaction (pp. 3-12). Florida, United States Springer, Cham.*

(5) - Publication d'un papier intitulée « **Emergent IoT Wireless Technologies beyond the year 2020 : A Comprehensive Comparative Analysis** » International Conference on Computer and Information Technology (ICCIT 2020), Tabuk, Saudi Arabia, 05/02/20-06/02/20, IEEE explore (Eds.)

*ABDALLAH Wajih, MNASRI Sami, VAL Thierry "Emergent IoT Wireless Technologies beyond the year 2020: A Comprehensive Comparative Analysis" International Conference on Computer and Information Technology (ICCIT 2020), Tabuk, Saudi Arabia, 05/02/20-06/02/20.*

(6) - Publication d'un papier intitulé « **Genetic-Voronoi algorithm for coverage of IoT data collection networks** » 30th International Conference on Computer Theory and Applications (ICCTA 2020). 12-14 December 2020. will be organized by the

Arab Academy for Science, Technology & Maritime Transport (AASTMT) in cooperation with the Computer Scientific Society (CSS) with technical co-sponsorship by IEEE Alexandria subsection.

[https://conferences.ieee.org/conferences\\_events/conferences/conferencedetails/52020](https://conferences.ieee.org/conferences_events/conferences/conferencedetails/52020)

*ABDALLAH, Wajih, MNASRI, Sami, et VAL, Thierry. Genetic-Voronoi algorithm for coverage of IoT data collection networks. In : 30th International Conference on Computer Theory and Applications (ICCTA 2020). 2020.*

<https://hal.archives-ouvertes.fr/hal-03039800/>

### **Conférences nationales**

(7) - Publication d'un papier intitulée « **Étude comparative des technologies sans fil utilisées pour l'IoT** » publiée dans les Journées Nationales des Communication terrestres, Colmar, France, 17/10/19-18/10/19.

*ABDALLAH Wajih, MNASRI Sami, VAL Thierry. “ Étude comparative des technologies sans fil utilisées pour l'IoT. Journées Nationales des Communications Terrestres (JNCT 2019), 17-18 oct. 2019 Colmar (France).*

## Bibliographie

- [1] Chong, C. Y., & Kumar, S. P. (2003). Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8), 1247-1256, doi:10.1109/jproc.2003.814918.
- [2] Bouckaert, S., et al. (2010). Strategies and challenges for interconnecting wireless mesh and wireless sensor networks. *Wireless Personal Communications*, 53(3), 443–463, doi: 10.1007/s11277-010-9957-z.
- [3] Chen, L., & Chen, L. (2014). An improved secure routing protocol based on clustering for wireless sensor networks, In *Mechatronics and automatic control systems*, Springer. pp. 995–1001, doi:10.1007/978-3-319-01273-5\_112.
- [4] Nguyen, S. T., Cayirci, E., & Rong, C. (2014). "A secure many-to-many routing protocol for wireless sensor and actuator networks". *Security and Communication Networks*, 7(1), 88–98, doi: 10.1002/sec.548.
- [5] Heragu, A., Ruffieux, D., & Enz, C. (2014). "The design of ultralow-power MEMS-based radio for WSN and WBAN". In *Frequency references, power management for SoC, and smart wireless interfaces*. Springer. pp. 265–280, doi : 10.1007/978-3-319-01080-9\_15.
- [6] Gupta, G. P., Misra, M., & Garg, K. (2014). Energy and trust aware mobile agent migration protocol for data aggregation in wireless sensor networks. *Journal of Network and Computer Applications*, 41, 300-311, doi: 10.1016/j.jnca.2014.01.003.
- [7] Chong, S. K., et al. (2011). "Energy-aware data processing techniques for wireless sensor networks: A review". In *Transactions on large-scale data-and knowledge-centered systems III*, Springer. pp. 117– 137, doi: 10.1007/978-3-642-23074-5\_5.
- [8] Chow, C.-Y., Xu, W., & He, T. (2014). "Privacy enhancing technologies for wireless sensor networks". *The Art of Wireless Sensor Networks*, v(1), 609–641, doi: 10.1007/978-3-642-40009-4\_18.
- [9] Kapnadak, V., & Coyle, E. J. (2014). "Optimal nonuniform deployment of sensors for distributed detection in wireless sensor networks". *ACM Transactions on Sensor Networks (TOSN)*, 10(2), 29, doi:10.1145/2530288.
- [10] Shaktawat, S. P. & Sharma, O. (2014). "Node deployment models and their performance parameters for wireless sensor network: A perspective". *International Journal of Computer Applications*, 88(9), doi: 10.1.1.429.294.
- [11] De Berg, M. et al., (2000). "Computational Geometry". In *Computational Geometry SE - 1*. Springer Berlin Heidelberg, pp. 1–17. Available at: [http://dx.doi.org/10.1007/978-3-662-04245-8\\_1](http://dx.doi.org/10.1007/978-3-662-04245-8_1).
- [12] Aziz, N., Aziz, K. & Ismail, W.,(2009). "Coverage strategies for wireless sensor networks". *World Academy of Science, Engineering and Technology*, 50, pp.145–150.
- [13] Argany, M. (2015). Development of a GIS-based method for sensor network deployment and coverage optimization.. Available at: <http://hdl.handle.net/20.500.11794/25829>



- [14] Priyadarshi, R., Gupta, B., & Anurag, A. (2020). "Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues". *The Journal of Supercomputing*, 76(9), 7333-7373, doi : 10.1007/s11227-020-03166-5.
- [15] Priyadarshi, R., Gupta, B., & Anurag, A. (2020)." Wireless sensor networks deployment: a result oriented analysis". *Wireless Personal Communications*, 113(2), 843-866, doi:10.1007/s11277-020-07255-9.
- [16] Aznoli, F., & Navimipour, N. J. (2017). Deployment strategies in the wireless sensor networks: systematic literature review, classification, and current trends. *Wireless Personal Communications*, 95(2), 819-846, doi: 10.1007/s11277-016-3800-0.
- [17] Farsi, M., Elhosseini, M. A., Badawy, M., Ali, H. A., & Eldin, H. Z. (2019). Deployment techniques in wireless sensor networks, coverage and connectivity: A survey. *Ieee Access*, 7, 28940-28954, doi: 10.1109/ACCESS.2019.2902072.
- [18] Mnasri, S. (2018). Contributions to the optimized deployment of connected sensors on the Internet of Things collection networks (Doctoral dissertation, Université Toulouse Jean Jaurès).
- [19] Kuila, P., & Jana, P. K. (2017). Clustering and routing algorithms for wireless sensor networks: energy efficient approaches (1st ed.). CRC Press (Taylor & Francis Group). ISBN-13: 978- 1498753821
- [20] Chowdary, V., & Bera, T. (2022). Types of Coverage in Wireless Sensor Network: A Survey. *Grenze International Journal of Engineering and Technology*, Jan Issue
- [21] Chowdary, V., Mondal, A. K., Sharma, A., & Capoor, S. (2021). Pre-Deployment Strategy for Maximizing Barrier Coverage in Wireless Sensor Network. *International Journal of Sensors Wireless Communications and Control*, 11(2), 181-188, doi: 10.2174/2210327910999200606230721
- [22] Chowdary, Vinay & Gupta, Mukul & Choudhury, Sushabhan. (2019). Maximizing Lifetime of Barrier Coverage Wireless Sensor Network with Tell Neighbor Algorithm. *International Journal of Mathematical, Engineering and Management Sciences*. 4. 775-785, doi:10.33889/IJMEMS.2019.4.3-060
- [23] Pinar, Y., Zuhair, A., Hamad, A., Resit, A., Shiva, K., & Omar, A. (2016, April). Wireless sensor networks (WSNs). In 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT) (pp. 1-8). IEEE, doi: 10.1109/LISAT.2016.7494144.
- [24] Yong-Min, L., Shu-Ci, W., & Xiao-Hong, N. (2009). The architecture and characteristics of wireless sensor network. In 2009 International Conference on Computer Technology and Development (Vol. 1, pp. 561-565). IEEE., doi: 10.1109/ICCTD.2009.44.
- [25] Wei Z, Jia Y, Yao Y, Zhu L, Guan L, Mao Y, Liu P, Zhang Y (2019) Discovering and understanding the security hazards in the interactions between IoT devices, mobile apps, and clouds on smart home platforms. In: 28th {USENIX} security symposium ({USENIX} security 19, pp 1133–1150
- [26] Morais, C. M. D., Sadok, D., & Kelner, J. (2019). An IoT sensor and scenario survey for data researchers. *Journal of the Brazilian Computer Society*, 25(1), 1-17, doi : 10.1186/s13173-019-0085-7.

- [27] Laghari, A. A., Wu, K., Laghari, R. A., Ali, M., & Khan, A. A. (2021). A review and state of art of Internet of Things (IoT). *Archives of Computational Methods in Engineering*, 1-19, doi:10.1007/s11831-021-09622-6
- [28]. Khelifi F, Bradai A, Benslimane A, Rawat P, Atri M (2019) A survey of localization systems in internet of things. *Mob Netw Appl* 24(3):761–785, doi:10.1007/s11036-018-1090-3.
- [29] Andrews, L. J. B., Raja, L., & Shanmugasundaram, S. (2019). Mobile android-based remote patient monitoring system through wearable sensors. *Journal of Discrete Mathematical Sciences and Cryptography*, 22(4), 557-568, doi:10.1080/09720529.2019.1637156.
- [30] Sachin B, Stango A, Prasad N, Sen J, Prasad R (2011) Proposed embedded security framework for internet of things (IoT). In: 2011 2nd international conference on wireless communication, vehicular technology, information theory and aerospace and electronic systems technology (wireless VITAE). IEEE, pp 1–5, doi: 10.1109/WIRELESSVITAE.2011.5940923.
- [31] Srivastava, S., Singh, M., & Gupta, S. (2018). Wireless sensor network: A survey. In 2018 international conference on automation and computational engineering, ICACE 2018, doi : 10.1109/ ICACE.2018.86870 59.
- [32] Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, doi : 10.1016/j.comnet.2008.04.002.
- [33] Xu, K., Hassanein, H., Takahara, G., & Wang, Q. (2010). Relay node deployment strategies in heterogeneous wireless sensor networks. *IEEE Transactions on Mobile Computing*. doi : 10.1109/ TMC.2009.105.
- [34] Liang, W., Luo, J., & Xu, X. (2010). Prolonging network lifetime via a controlled mobile sink in wireless sensor networks. *GLOBECOM—IEEE Global Telecommunications Conference*, doi : 10.1109/GLOCOM.2010.56830 95.
- [35] Ren, J., Zhang, Y., Zhang, K., Liu, A., Chen, J., & Shen, X. S. (2016). Lifetime and energy hole evolution analysis in data-gathering wireless sensor networks. *IEEE Transactions on Industrial Informatics*, doi : 10.1109/TII.2015.24112 31.
- [36] Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer networks*, 52(12), 2292-2330. doi : 10.1016/j.comnet.2008.04.002
- [37] Rawat P, Singh KD, Chaouchi H, Bonnin JM (2014) Wireless sensor networks: a survey on recent developments and potential synergies. *J Supercomput* 68:1–48. doi:10.1007/s11227-013-1021-9.
- [38] Zheng J, Jamalipour A (2009) *Wireless sensor networks: a networking perspective*. John Wiley & Sons
- [39] Dulman SO, Chatterjea S, Havinga PJ (2005) Introduction to wireless sensor networks. In: *Embedded systems handbook*. Taylor & Francis CRC Press, pp 3-1 ISBN 9781315218298

- [40] Pereira F, Correia R, Carvalho NB (2018) Comparison of active and passive sensors for IoT applications. In: 2018 IEEE Wireless Power Transfer Conference (WPTC), pp 1–3. Doi : 10.1109/WPT.2018.8639445
- [41] Akan O, Talha Isik M, Baykal B (2009) Wireless passive sensor networks. in IEEE Communications Magazine, vol. 47, no. 8, pp. 92-99,. doi: 10.1109/MCOM.2009.5181898
- [42] Wang Y, Zhang Y, Liu J, Bhandari R (2015). Coverage, connectivity, and deployment in wireless sensor networks. Recent development in wireless sensor and ad-hoc networks. Springer, New Delhi, pp 25–44. DOI: 10.1007/978-81-322-2129-6\_2
- [43] Doraipandian, M., & Neelamegam, P. (2013). Wireless sensor network using ARM processors: a review in hardware perspective. International Journal of Embedded and Real-Time Communication Systems (IJERTCS), 4(4), 48-59. DOI: 10.4018/ijertcs.2013100103
- [44] Gkikopouli, A., Nikolakopoulos, G., & Manesis, S. (2012, July). A survey on underwater wireless sensor networks and applications. In 2012 20th Mediterranean conference on control & automation (MED) (pp. 1147-1154). IEEE. doi: 10.1109/MED.2012.6265793.
- [45] Nadeem, F., Leitgeb, E., Awan, M. S., & Chessa, S. (2009, August). Comparing the life time of terrestrial wireless sensor networks by employing hybrid FSO/RF and only RF access networks. In 2009 Fifth International Conference on Wireless and Mobile Communications (pp. 134-139). IEEE. doi: 10.1109/ICWMC.2009.29.
- [46] Tariq, A. K., Ziyad, A. T., & Abdullah, A. O. (2013). Wireless sensor networks for leakage detection in underground pipelines: a survey paper. Procedia Computer Science, 21, 491-498. <https://doi.org/10.1016/j.procs.2013.09.067>
- [47] Sun, Z. H. I., Akyildiz, I. F., & Hancke, G. P. (2011). Dynamic connectivity in wireless underground sensor networks. IEEE Transactions on Wireless Communications, 10(12), 4334-4344. doi: 10.1109/TWC.2011.093011.110632
- [48] Ayaz, M., Baig, I., Abdullah, A., & Faye, I. (2011). A survey on routing techniques in underwater wireless sensor networks. Journal of Network and Computer Applications, 34(6), 1908-1927. doi : 10.1016/j.jnca.2011.06.009
- [49] Anguita, D., Brizzolaro, D., & Parodi, G. (2009, June). Building an underwater wireless sensor network based on optical: Communication: Research challenges and current results. In 2009 Third International Conference on Sensor Technologies and Applications (pp. 476-479). IEEE. doi: 10.1109/SENSORCOMM.2009.79.
- [50] Tonneau, A. S., Mitton, N., & Vandaele, J. (2014, May). A survey on (mobile) wireless sensor network experimentation testbeds. In 2014 IEEE International Conference on Distributed Computing in Sensor Systems (pp. 263-268). IEEE. doi: 10.1109/DCOSS.2014.41
- [51] Shen, H., & Bai, G. (2016). Routing in wireless multimedia sensor networks: A survey and challenges ahead. Journal of Network and Computer Applications, 71, 30-49. Doi : 10.1016/j.jnca.2016.05.013

- [52] Akyildiz, I. F., Melodia, T., & Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks. *Computer networks*, 51(4), 921-960. Doi : 10.1016/j.comnet.2006.10.002
- [53] **Abdallah, W.**, Mnasri, S., Nasri, N., & Val, T. (2020). Emergent IoT wireless technologies beyond the year 2020: A comprehensive comparative analysis. In 2020 International Conference on Computing and Information Technology (ICCIT-1441) (pp. 1-5). IEEE. doi: 10.1109/ICCIT-144147971.2020.9213799.
- [54] **Abdallah, W.**, Mnasri, S., & Val, T. (2019). Étude comparative des technologies sans fil utilisées pour l'IoT. In Journées Nationales des Communications Terrestres (JNCT 2019).
- [55] LoRa world coverage: [www.lora-alliance.org/](http://www.lora-alliance.org/) (consulté le 10/05/2022)
- [56] F. Sforza, Communications system, 26 March 2013, US Patent US8406275 B2. ~~https://patents.google.com/patent/US8406275~~ »
- [57] Reynders, B., Meert, W., & Pollin, S. (2016, May). Range and coexistence analysis of long range unlicensed communication. In 2016 23rd International Conference on Telecommunications (ICT) (pp. 1-6). IEEE, doi: 10.1109/ICT.2016.7500415.
- [58] Mikhaylov, K., Petaejaejaervi, J., & Haenninen, T. (2016, May). Analysis of capacity and scalability of the LoRa low power wide area network technology. In European Wireless 2016; 22th European Wireless Conference (pp. 1-6). VDE.
- [59] LoRa Alliance, <https://www.lora-alliance.org/lorawan-for-developers>, LoRaWAN Specification, 2016 ed.
- [60] Ayoub, W., Samhat, A. E., Nouvel, F., Mroue, M., & Prévotet, J. C. (2018). Internet of mobile things: Overview of lorawan, dash7, and nb-iot in lpwans standards and supported mobility. *IEEE Communications Surveys & Tutorials*, 21(2), 1561-1581, doi: 10.1109/COMST.2018.2877382.
- [61] Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2019). A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express*, 5(1), 1-7, doi: 10.1016/j.icte.2017.12.005
- [62] Sigfox world coverage : [www.sigfox.com/en/coverage/](http://www.sigfox.com/en/coverage/)
- [63] El Mahjoubi, A., Mazri, T., & Hmina, N. (2017, November). M2M and eMTC communications via NB-IoT, Morocco first testbed experimental results and RF deployment scenario: New approach to improve main 5G KPIs and performances. In 2017 International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 1-6). IEEE, doi: 10.1109/WINCOM.2017.8238156.
- [64] Nair, K. K., Abu-Mahfouz, A. M., & Lefophane, S. (2019, March). Analysis of the narrow band internet of things (NB-IoT) technology. In 2019 conference on information communications technology and society (ICTAS) (pp. 1-6). IEEE, doi: 10.1109/ICTAS.2019.8703630.
- [65] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE*

communications surveys & tutorials, 17(4), 2347-2376, doi: 10.1109/COMST.2015.2444095.

[66] Bluetooth, S. I. G. (2010). Specification of the bluetooth system-covered core package version: 4.0. Bluetooth Special Interest Group.

[67] Bluetooth, S. I. G. (2014). Specification of the bluetooth system v4. 2. Standard, Bluetooth SIG, 27.

[68] Yaakop, M. B., Abd Malik, I. A., bin Suboh, Z., Ramli, A. F., & Abu, M. A. (2017, September). Bluetooth 5.0 throughput comparison for internet of thing usability a survey. In 2017 International Conference on Engineering Technology and Technopreneurship (ICE2T) (pp. 1-6). IEEE, doi: 10.1109/ICE2T.2017.8215995.

[69] Bluetooth 5 Learning. Available online : <https://www.silabs.com/products/wireless/learning-center/bluetooth/bluetooth-5> (accessed on 3 July 2019).

[70] Sun, D. Z., Sun, L., & Yang, Y. (2019). On secure simple pairing in Bluetooth standard v5. 0-part II: privacy analysis and enhancement for low energy. *Sensors*, 19(15), 3259, doi: 10.3390/s19153259

[71] Yu, Y., Ruan, F., Li, C., Jiang, T., Tang, C., & Wu, H. (2013). Environment green energy harvesting technology applied in wireless sensor networks. *Journal of Network & Information Security*, 4(4), 361-367.

[72] Gazis, V., Görtz, M., Huber, M., et al., E. (2015, August). A survey of technologies for the internet of things. In 2015 international wireless communications and mobile computing conference (IWCMC) (pp. 1090-1095). IEEE. doi: 10.1109/IWCMC.2015.7289234.

[73] FRANCOIS, E. EnOcean: Un standard universel pour le Smart Home & le Smart Building IBS-09/2013.

[74] Iannello, F., Simeone, O., & Spagnolini, U. (2012). Medium access control protocols for wireless sensor networks with energy harvesting. *IEEE Transactions on Communications*, 60(5), 1381-1389, doi: 10.1109/TCOMM.2012.030712.110089

[75] Ploennigs, J., Ryssel, U., & Kabitzsch, K. (2010, September). Performance analysis of the EnOcean wireless sensor network protocol. In 2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010) (pp. 1-9). IEEE, doi: 10.1109/ETFA.2010.5641313.

[76] Nikoogar, H., & Prasad, R. (2009). UWB regulation. In *Introduction to Ultra Wideband for Wireless Communications* (pp. 163-171). Springer, Dordrecht. ISBN : 978-1-4020-6632-0

[77] Minoli, D., & Occhiogrosso, B. (2018, September). Ultrawideband (UWB) technology for smart cities IoT applications. In 2018 IEEE international smart cities conference (ISC2) (pp. 1-8). IEEE, doi: 10.1109/ISC2.2018.8656958.

[78] Fofana, N. I., Van den Bossche, A., & Val, T. (2015). Un banc de mesure de ranging pour la localisation par ToF de noeuds UWB pour les Réseaux de Capteurs Sans Fil et l'Internet des Objets. In: 5emes Journées Nationales des Communications Terrestres (JNCT 2015), 4 June 2015 - 5 June 2015 (Valence, France)

- [79] Sharma, H., & Sharma, S. (2014). A review of sensor networks: Technologies and applications. 2014 Recent Advances in Engineering and Computational Sciences (RAECS), 1-4, doi: 10.1109/RAECS.2014.6799579.
- [80] Li, L., Xiaoguang, H., Jian, H., & Ketai, H. (2011, June). Design of new architecture of AMR system in Smart Grid. In 2011 6th IEEE Conference on Industrial Electronics and Applications (pp. 2025-2029). IEEE, doi: 10.1109/ICIEA.2011.5975925.
- [81] Jiang, X., Zhang, H., Yi, E. A. B., Raghunathan, N., Mousoulis, C., Chaterji, (2020). Hybrid low-power wide-area mesh network for iot applications. IEEE Internet of Things Journal, 8(2), 901-915, doi: 10.1109/JIOT.2020.3009228.
- [82] Liu, L., Xia, F., Wang, Z., Chen, J., & Sun, Y. (2005, December). Deployment issues in wireless sensor networks. In International conference on mobile ad-hoc and sensor networks (pp. 239-248). Springer, Berlin, Heidelberg, doi : 10.1007/11599463\_24
- [83] Navimipour, N. J., & Rahmani, A. M. (2009, November). The new genetic based method with optimum number of super node in heterogeneous wireless sensor network for fault tolerant system. In 2009 International Conference on Intelligent Networking and Collaborative Systems (pp. 187-193). IEEE, doi: 10.1109/INCOS.2009.86.
- [84] Deif, D. S., & Gadallah, Y. (2014). Classification of wireless sensor networks deployment techniques. IEEE Communications Surveys & Tutorials, 16(2), 834–855, doi: 10.1109/SURV.2013.091213.00018.
- [85] Hossain A, Biswas PK, Chakrabarti S (2008) 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, pp 1–5, doi: 10.1109/ICIINFS.2008.4798455.
- [86] Wang, Z., Bulut, E., & Szymanski, B. K. (2008, December). Distributed target tracking with imperfect binary sensor networks. In IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference (pp. 1-5). IEEE, doi: 10.1109/GLOCOM.2008.ECP.62.
- [87] Abdollahzadeh, S., & Navimipour, N. J. (2016). Deployment strategies in the wireless sensor network: A comprehensive review. Computer Communications, 91, 1-16, doi : 10.1016/j.comcom.2016.06.003
- [88] Zou, Y., & Chakrabarty, K. (2003, March). Sensor deployment and target localization based on virtual forces. In IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428) (Vol. 2, pp. 1293-1303), doi: 10.1109/INFCOM.2003.1208965.
- [89] Liu, B., Dousse, O., Nain, P., & Towsley, D. (2012). Dynamic coverage of mobile sensor networks. IEEE Transactions on Parallel and Distributed systems, 24(2), 301-311, doi: 10.1109/TPDS.2012.141.
- [90] Liu, L., Xia, F., Wang, Z., Chen, J., & Sun, Y. (2005, December). Deployment issues in wireless sensor networks. In International conference on mobile ad-hoc and sensor networks (pp. 239-248). Springer, Berlin, Heidelberg, doi : 10.1007/11599463\_24
- [91] Song, C., Liu, L., Feng, G., & Fan, Y. (2013, May). Persistent awareness coverage with maximum coverage frequency for mobile sensor networks. In 2013 IEEE International

Conference on Cyber Technology in Automation, Control and Intelligent Systems (pp. 235-240). IEEE, doi: 10.1109/CYBER.2013.6705451.

[92] Khanjary, M., Sabaei, M., & Meybodi, M. R. (2015). Critical density for coverage and connectivity in two-dimensional fixed-orientation directional sensor networks using continuum percolation. *Journal of Network and Computer Applications*, 57, 169-181, doi : 10.1016/j.jnca.2015.08.010

[93] Song, C., Liu, L., Feng, G., & Fan, Y. (2013, May). Persistent awareness coverage with maximum coverage frequency for mobile sensor networks. In 2013 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems (pp. 235-240), doi: 10.1109/CYBER.2013.6705451.

[94] Gupta, H. P., Rao, S. V., & Venkatesh, T. (2016). Analysis of stochastic coverage and connectivity in three-dimensional heterogeneous directional wireless sensor networks. *Pervasive and Mobile Computing*, 29, 38-56, doi : 10.1016/j.pmcj.2015.08.004

[95] Patzer, A., & Patzer, A. (2002). Deployment techniques. *JSP Examples and Best Practices*. [https://doi.org/10.1007/978-1-4302-0831-0\\_10](https://doi.org/10.1007/978-1-4302-0831-0_10).

[96] Beutel, J., Römer, K., Ringwald, M., & Woehrle, M. (2010). Deployment techniques for sensor. *Networks*. [https://doi.org/10.1007/978-3-642-01341-6\\_9](https://doi.org/10.1007/978-3-642-01341-6_9)

[97] Abidin, H. Z., Din, N. M., Radzi, N. A. M., & Rizman, Z. I. (2017). A review on sensor node placement techniques in wireless sensor networks. *International Journal on Advanced Science, Engineering and Information Technology*, 7(1), 190-197, ISSN: 2088-5334

[98] Wang X, Wang S (2011) Hierarchical deployment optimization for wireless sensor networks. *IEEE Transactions on Mobile Computing*, 10(7), 1028-1041, doi : 10.1109/TMC.2010.216.

[99] Mougou, K., Mahfoudh, S., Minet, P., & Laouiti, A. (2012, September). Redeployment of randomly deployed wireless mobile sensor nodes. In 2012 IEEE Vehicular Technology Conference (VTC Fall) (pp. 1-5). IEEE, doi: 10.1109/VTCFall.2012.6399012.

[100] Yu, X., Liu, N., Huang, W., Qian, X., & Zhang, T. (2013). A node deployment algorithm based on van der Waals force in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 9(10), 505710, doi : 10.1155/2013/505710

[101] Lee, D. T., & Schachter, B. J. (1980). Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3), 219-242, doi : 10.1007/BF00977785

[102] Al-Turjman, F., Hassanein, H. S., & Ibnkahla, M. (2017). Quantifying Connectivity in Wireless Sensor Networks with Grid-Based Deployments 1. In *Cognitive Sensors and IoT* (pp. 77-106). CRC Press. ISBN 9781315103600

[103] Kim, Y. H., Kim, C. M., Yang, D. S., Oh, Y. J., & Han, Y. H. (2012, February). Regular sensor deployment patterns for p-coverage and q-connectivity in wireless sensor networks. In *The International Conference on Information Network 2012* (pp. 290-295). IEEE, doi: 10.1109/ICOIN.2012.6164394.

- [104] Liu, Y., Suo, L., Sun, D., & Wang, A. (2013). A virtual square grid-based coverage algorithm of redundant node for wireless sensor network. *Journal of network and computer applications*, 36(2), 811-817, doi : 10.1016/j.jnca.2012.12.003
- [105] Kumari, P., & Singh, Y. (2012, January). Delaunay triangulation coverage strategy for wireless sensor networks. In *2012 International Conference on Computer Communication and Informatics* (pp. 1-5), doi: 10.1109/ICCCI.2012.6158874.
- [106] Maksimović, M., & Milošević, V. (2016). Evaluating the optimal sensor placement for smoke detection. *Yugoslav journal of operations research*, 26(1), 33-50, doi : 10.2298/YJOR140312002M
- [107] Liu, Y., Suo, L., Sun, D., & Wang, A. (2013). A virtual square grid-based coverage algorithm of redundant node for wireless sensor network. *Journal of network and computer applications*, 36(2), 811-817, doi : 10.1016/j.jnca.2012.12.003
- [108] Kim, Y. H., Kim, C. M., Yang, D. S., Oh, Y. J., & Han, Y. H. (2012, February). Regular sensor deployment patterns for p-coverage and q-connectivity in wireless sensor networks. In *The International Conference on Information Network 2012* (pp. 290-295), doi: 10.1109/ICOIN.2012.6164394.
- [109] O'Rourke, J. (1998). *Computational Geometry in C* (2nd ed.). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511804120
- [110] Fortune, S. (1995). Voronoi diagrams and Delaunay triangulations. *Computing in Euclidean geometry*, 225-265, doi : 10.1142/9789812831699\_0007
- [111] Sung, T. W., & Yang, C. S. (2014). Voronoi-based coverage improvement approach for wireless directional sensor networks. *Journal of Network and Computer Applications*, 39, 202-213, doi : 10.1016/j.jnca.2013.07.003
- [112] Nguyen, H. T., & Le, H. X. (2016). Path planning and obstacle avoidance approaches for mobile robot. *arXiv preprint arXiv:1609.01935*, doi : 10.48550/arXiv.1609.01935
- [113] Li, Y., Dong, T., Bikdash, M., & Song, Y. D. (2005, June). Path Planning for Unmanned Vehicles Using Ant Colony Optimization on a Dynamic Voronoi Diagram. In *IC-AI* (pp. 716-721)..
- [114] Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life reviews*, 2(4), 353-373, doi : 10.1016/j.plrev.2005.10.001
- [115] Wu, C. H., Lee, K. C., & Chung, Y. C. (2007). A Delaunay triangulation based method for wireless sensor network deployment. *Computer Communications*, 30(14-15), 2744-2752, doi : 10.1016/j.comcom.2007.05.017
- [116] Lloyd, E. L. (1977, October). On triangulations of a set of points in the plane. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)* (pp. 228-240), doi: 10.1109/SFCS.1977.21.
- [117] Sarkar, R., Yin, X., Gao, J., Luo, F., & Gu, X. D. (2009, April). Greedy routing with guaranteed delivery using ricci flows. In *2009 International Conference on Information Processing in Sensor Networks* (pp. 121-132). IEEE..



- [118] Zhou, H., Wu, H., Xia, S., Jin, M., & Ding, N. (2011, April). A distributed triangulation algorithm for wireless sensor networks on 2D and 3D surface. In 2011 Proceedings IEEE INFOCOM (pp. 1053-1061), doi: 10.1109/INFOCOM.2011.5934879.
- [119] Zhou, H., Xia, S., Jin, M., & Wu, H. (2010, June). Localized algorithm for precise boundary detection in 3D wireless networks. In 2010 IEEE 30th International Conference on Distributed Computing Systems (pp. 744-753), doi: 10.1109/ICDCS.2010.40.
- [120] Zhou, H., Jin, M., & Wu, H. (2013, July). A distributed Delaunay triangulation algorithm based on centroidal Voronoi tessellation for wireless sensor networks. In Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing (pp. 59-68), doi: 10.1145/2491288.2491296
- [121] De Berg, M., Cheong, O., Van Kreveld, M., & Overmars, M. (2008). Computational geometry: introduction. Computational geometry: algorithms and applications, 1-17, doi: 10.1007/978-3-540-77974-2\_1
- [122] Lederer, S., Wang, Y., & Gao, J. (2009). Connectivity-based localization of large-scale sensor networks with complex shape. ACM Transactions on Sensor Networks (TOSN), 5(4), 1-32, doi: 10.1145/1614379.1614383
- [123] Jin, M., Rong, G., Wu, H., Shuai, L., & Guo, X. (2012, March). Optimal surface deployment problem in wireless sensor networks. In 2012 Proceedings IEEE INFOCOM (pp. 2345-2353), doi: 10.1109/INFOCOM.2012.6195622.
- [124] Dr'eo, J. et al., (2006). Metaheuristics for Hard Optimization, Berlin, Germany: Springer-Verlag. ISBN-10 3-540-23022-X Springer Berlin Heidelberg New York
- [125] Haupt, R.L., Haupt, S.E. & Wiley, A.J.,(2004). ALGORITHMS PRACTICAL GENETIC ALGORITHMS 2nd ed., Hoboken, NJ: John Wiley & Sons.ISBN-0-471-45565-2
- [126] Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press. Online ISBN: 9780262275552.
- [127] Mnasri, S., Nasri, N., van den Bossche, A., & Thierry, V. A. L. (2018, June). 3D indoor redeployment in IoT collection networks: a real prototyping using a hybrid PI-NSGA-III-VF. In 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC) (pp. 780-785). IEEE., pp. 780-785, doi: 10.1109/IWCMC.2018.8450372.
- [128] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948), doi: 10.1109/ICNN.1995.488968.
- [129] Li, Y., & Cao, J. (2021). WSN node optimal deployment algorithm based on adaptive binary particle swarm optimization. ASP Transactions on Internet of Things, 1(1), 1-8, doi: 10.52810/TIOT.2021.100026
- [130] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200, pp. 1-10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department..

- [131] Dorigo, M., Maniezzo, V., & Coloni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1), 29-41, doi: 10.1109/3477.484436.
- [132] Pan, W. (2012). A new fruit fly optimization algorithm: Taking the financial distress model as an example. *Knowledge-Based Systems*, 26, 69–74, doi: 10.1016/j.knsys.2011.07.001
- [133] Passino, K. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems*, 22(3), 52–67, doi: 10.1109/MCS.2002.1004010.
- [134] He, S., Wu, Q. H., & Saunders, J. R. (2006, July). A novel group search optimizer inspired by animal behavioural ecology. In *2006 IEEE international conference on evolutionary computation* (pp. 1272-1278), doi: 10.1109/CEC.2006.1688455.
- [135] El-Abd, M. (2013). An improved global-best harmony search algorithm. *Applied Mathematics and Computation*, 222, 94–106, doi: 10.1016/j.amc.2013.07.020
- [136] Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: Charged system search. *Acta Mechanica*, 213(3), 267–289, doi: 10.1007/s00707-009-0270-4
- [137] Rabanal, P., Rodríguez, I., & Rubio, F. (2007, August). Using river formation dynamics to design heuristic algorithms. In *International conference on unconventional computation* (pp. 163-177). Springer, Berlin, Heidelberg, doi: 10.1007/978-3-540-73554-0\_16
- [138] Gazen, C., & Ersoy, C. (1999). Genetic algorithms for designing multihop lightwave network topologies. *Artificial Intelligence in Engineering*, 13(3), 211-221, doi: 10.1016/S0954-1810(98)00019-3
- [139] Grefenstette, J. J. (1993, August). Genetic algorithms and machine learning. In *Proceedings of the sixth annual conference on Computational learning theory* (pp. 3-4). ACM 0-89791-611-5/9310003
- [140] Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994, June). A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence* (pp. 82-87), doi: 10.1109/ICEC.1994.350037.
- [141] CHEN, H. C., & HIRASAWA, K. (2006). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, 1975. *IEICE transactions on information and systems*, 89(12), 3012-3014.
- [142] Konstantinidis, A., & Yang, K. (2011). Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific MOEA/D. *Applied Soft Computing*, 11(6), 4117-4134, doi: 10.1016/j.asoc.2011.02.031
- [143] Konstantinidis, A., & Yang, K. (2011). Multi-objective k-connected deployment and power assignment in wsns using a problem-specific constrained evolutionary algorithm based on decomposition. *Computer Communications*, 34(1), 83-98, doi: 10.1016/j.comcom.2010.08.011

- [144] Konstantinidis, A., Yang, K., & Zhang, Q. (2008, November). An evolutionary algorithm to a multi-objective deployment and power assignment problem in wireless sensor networks. In *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference* (pp. 1-6), doi: 10.1109/GLOCOM.2008.ECP.98.
- [145] Qu, Y., & Georgakopoulos, S. V. (2011, April). Relocation of wireless sensor network nodes using a genetic algorithm. In *WAMICON 2011 Conference Proceedings* (pp. 1-5), doi: 10.1109/WAMICON.2011.5872882.
- [146] Rahmani, N., Nematy, F., Rahmani, A. M., & Hosseinzadeh, M. (2011). Node placement for maximum coverage based on voronoi diagram using genetic algorithm in wireless sensor networks. *Australian Journal of Basic and Applied Sciences*, 5(12), 3221-3232.
- [147] Xu, Y., & Yao, X. (2006, January). A GA approach to the optimal placement of sensors in wireless sensor networks with obstacles and preferences. In *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.* (Vol. 1, pp. 127-131), doi: 10.1109/CCNC.2006.1593001.
- [148] Kalayci, T. E., & Uğur, A. (2011). Genetic algorithm–based sensor deployment with area priority. *Cybernetics and Systems*, 42(8), 605-620, doi: 10.1080/01969722.2011.634676
- [149] Wang, X., Wang, S., & Jiang, A. (2006, October). Optimized deployment strategy of mobile agents in wireless sensor networks. In *Sixth International Conference on Intelligent Systems Design and Applications* (Vol. 2, pp. 893-898), doi: 10.1109/ISDA.2006.253730.
- [150] Banimelhem, O., Mowafi, M., & Aljoby, W. (2013). Genetic algorithm based node deployment in hybrid wireless sensor networks. *Communications and Network*, 2013, doi:10.4236/cn.2013.54034
- [151] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200, pp. 1-10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- [152] Riley, J. R., Greggers, U., Smith, A. D., Reynolds, D. R., & Menzel, R. (2005). The flight paths of honeybees recruited by the waggle dance. *Nature*, 435(7039), 205-207, doi: 10.1038/nature03526
- [153] Karaboga, N. (2009). A new design method based on artificial bee colony algorithm for digital IIR filters. *Journal of the Franklin Institute*, 346(4), 328-348, doi: 10.1016/j.jfranklin.2008.11.003
- [154] Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied soft computing*, 8(1), 687-697, doi: 10.1016/j.asoc.2007.05.007
- [155] Karaboga, D., & Akay, B. (2009). A survey: algorithms simulating bee swarm intelligence. *Artificial intelligence review*, 31(1), 61-85, doi: 10.1007/s10462-009-9127-4.
- [156] Karaboga, D., Akay, B., & Ozturk, C. (2007, August). Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. In *International conference on modeling decisions for artificial intelligence* (pp. 318-329). Springer, Berlin, Heidelberg, doi: 10.1007/978-3-540-73729-2\_30

- [157] Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2014). A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*, 42(1), 21-57, doi.org/10.1007/s10462-012-9328-0
- [158] Öztürk, C., Karaboğa, D., & GÖRKEMLİ, B. (2012). Artificial bee colony algorithm for dynamic deployment of wireless sensor networks. *Turkish Journal of Electrical Engineering & Computer Sciences*, 20(2), 255-262, doi:10.3906/elk-1101-1030.
- [159] Udgata, S. K., Sabat, S. L., & Mini, S. (2009, December). Sensor deployment in irregular terrain using artificial bee colony algorithm. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)* (pp. 1309-1314), doi: 10.1109/NABIC.2009.5393734.
- [160] Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the sixth international symposium on micro machine and human science* (pp. 39-43), doi: 10.1109/MHS.1995.494215.
- [161] Fan, Z., & Zhao, W. (2011). Network coverage optimization strategy in wireless sensor networks based on particle swarm optimization.
- [162] Dorigo M, Blum C (2005) Ant colony optimization theory: a survey. *Theor Comput Sci* 344(2–3):243–278, doi: 10.1016/j.tcs.2005.05.020
- [163] Liao, W. H., Kao, Y., & Wu, R. T. (2011). Ant colony optimization based sensor deployment protocol for wireless sensor networks. *Expert Systems with Applications*, 38(6), 6599-6605, doi: 10.1016/j.eswa.2010.11.079
- [164] Du, K. L., & Swamy, M. N. S. (2016). Search and optimization by metaheuristics. *Techniques and Algorithms Inspired by Nature*, 1-10. Birkhauser, Basel, Switzerland, ISBN: 978-3-319-41192-7
- [165] El Khamlichi, Y., Tahiri, A., Abtoy, A., Medina-Bulo, I., & Palomo-Lozano, F. (2017). A hybrid algorithm for optimal wireless sensor network deployment with the minimum number of sensor nodes. *Algorithms*, 10(3), 80, doi: 10.3390/a10030080
- [166] Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: a new bio-inspired optimization algorithm. *Communications in nonlinear science and numerical simulation*, 17(12), 4831-4845, doi: 10.1016/j.cnsns.2012.05.010
- [167] Wang GG, Gandomi AH, Alavi AH (2014) Stud krill herd algorithm. *Neurocomputing* 128:363–370, doi: 10.1016/j.neucom.2013.08.031
- [168] Moh'd Alia, O., & Al-Ajouri, A. (2016). Maximizing wireless sensor network coverage with minimum cost using harmony search algorithm. *IEEE Sensors Journal*, 17(3), 882-896, doi: 10.1109/JSEN.2016.2633409.
- [169] Pietrabissa, A., Liberati, F., & Oddi, G. (2016). A distributed algorithm for Ad-hoc network partitioning based on Voronoi Tessellation. *Ad Hoc Networks*, 46, 37-47, doi: 10.1016/j.adhoc.2016.03.008
- [170] Banimelhem, O., Mowafi, M., & Aljoby, W. (2013). Genetic algorithm based node deployment in hybrid wireless sensor networks. *Communications and Network*, 2013, doi:10.4236/cn.2013.54034.

- [171] Eledlebi, K., Ruta, D., Saffre, F., Al-Hammadi, Y., & Isakovic, A. F. (2018, July). Autonomous deployment of mobile sensors network in an unknown indoor environment with obstacles. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 280-281), doi: 10.1145/3205651.3205725.
- [172] Eledlebi, K., Ruta, D., Saffre, F., AlHammadi, Y., & Isakovic, A. F. (2018, September). Voronoi-based indoor deployment of mobile sensors network with obstacles. In *2018 IEEE 3rd International Workshops on Foundations and Applications of Self\* Systems (FAS\* W)* (pp. 20-21). IEEE, doi: 10.1109/FAS-W.2018.00019.
- [173] Zou, J., Gundry, S., Kusyk, J., Sahin, C. S., & Uyar, M. Ü. (2014, December). Bio-inspired and Voronoi-based algorithms for self-positioning of autonomous vehicles in noisy environments. In *Proceedings of the 8th International Conference on Bioinspired Information and Communications Technologies* (pp. 17-22), doi: 10.4108/icst.bict.2014.257917.
- [174] Eledlebi, K., Hildmann, H., Ruta, D., & Isakovic, A. F. (2020). A hybrid Voronoi tessellation/genetic algorithm approach for the deployment of drone-based nodes of a self-organizing wireless sensor network (WSN) in unknown and GPS denied environments. *Drones*, 4(3), 33, doi: 10.3390/drones4030033
- [175] Tahir, N. H. M., & Atan, F. (2016). A Modified Genetic Algorithm Method for Maximum Coverage in Dynamic Mobile Wireless Sensor Networks. *J. Basic Appl. Sci. Res*, 6, 26-32. ISSN 2090-430
- [176] Li, Y., Dong, T., Bikdash, M., & Song, Y. D. (2005, June). Path Planning for Unmanned Vehicles Using Ant Colony Optimization on a Dynamic Voronoi Diagram. In *IC-AI* (pp. 716-721), doi: 10.1.1.128.669
- [177] Ab Aziz, N. A. B., Moheemmed, A. W., & Alias, M. Y. (2009, March). A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram. In *2009 international conference on networking, sensing and control* (pp. 602-607), doi: 10.1109/ICNSC.2009.4919346.
- [178] Nematy, F., & Rahmani, N. (2013). Using Voronoi diagram and genetic algorithm to deploy nodes in wireless sensor network. *Intl. J. of Soft Computing and Software Engineering [JSCSE]*, 3(3), 706-713, doi: 10.7321/jscse.v3.n3.107.
- [179] Jog, P., Suh, J. Y., & Van Gucht, D. (1991). Parallel genetic algorithms applied to the traveling salesman problem. *SIAM Journal on Optimization*, 1(4), 515-529, doi: 10.1137/0801031
- [180] Goldberg, D. E. (1989, June). Sizing populations for serial and parallel genetic algorithms. In *Proceedings of the 3rd international conference on genetic algorithms* (pp. 70-79), doi: 10.5555/645512.657266
- [181] Cantú-Paz, E. (1998). A survey of parallel genetic algorithms. *Calculateurs paralleles, reseaux et systems repartis*, 10(2), 141-171, doi=10.1.1.588.6072
- [182] Di Martino, S., Ferrucci, F., Maggio, V., & Sarro, F. (2013). Towards migrating genetic algorithms for test data generation to the cloud. In *Software Testing in the Cloud*:

Perspectives on an Emerging Discipline (pp. 113-135). IGI Global, DOI: 10.4018/978-1-4666-2536-5.ch006

[183] Di Geronimo, L., Ferrucci, F., Murolo, A., & Sarro, F. (2012, April). A parallel genetic algorithm based on hadoop mapreduce for the automatic generation of junit test suites. In 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation (pp. 785-793), doi: 10.1109/ICST.2012.177.

[184] Herrera, F. & Lozano, M. (2000). Gradual distributed real-Coded genetic algorithms. IEEE Transactions on Evolutionary Computation 4(1), 43–63, doi: 10.1109/4235.843494.

[185] Yu, W., & Zhang, W. (2006, November). Study on function optimization based on master-slave structure genetic algorithm. In 2006 8th international Conference on Signal Processing (Vol. 3), doi: 10.1109/ICOSP.2006.345926.

[186] Gong, Y.J., Chen, W.N., Zhan, Z.H., Zhang, J., Li, Y., Zhang, Q., & Li, J.J. (2015). Distributed evolutionary algorithm and their models: A survey of state-of-the-art. Applied Soft Computing, 34, 286-300, doi: 10.1016/j.asoc.2015.04.061

[187] Muttillio, V., Giammatteo, P., Fiorilli, G., & Pomante, L. (2020, January). An OpenMP Parallel Genetic Algorithm for Design Space Exploration of Heterogeneous Multi-processor Embedded Systems. In Proceedings of the 11th Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures/9th Workshop on Design Tools and Architectures for Multicore Embedded Computing Platforms, New York, NY, United States,(pp. 1-6), doi: 10.1145/3381427.3381431

[188] Shen, C., Cheng, W., Liao, X., & Peng, S. (2008, May). Barrier coverage with mobile sensors. In 2008 International Symposium on Parallel Architectures, Algorithms, and Networks (i-span 2008) (pp. 99-104). IEEE, doi: 10.1109/I-SPAN.2008.8.

[189] Ban, D., Yang, W., Jiang, J., Wen, J., & Dou, W. (2010). Energy-efficient algorithms for k-barrier coverage in mobile sensor networks. International Journal of Computers Communications & Control, 5(5), 616-624. ISSN 1841-9836, E-ISSN 1841-9844

[190] Ahmed, N., Kanhere, S. S., & Jha, S. (17/11/2005). Probabilistic coverage in wireless sensor networks. In The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05), Sydney, NSW, Australia, 1 (pp. 8-pp). IEEE, doi: 10.1109/LCN.2005.109.

[191] Wu, J., Cardei, M., Dai, F., & Yang, S. (2006). Extended dominating set and its applications in ad hoc networks using cooperative communication. IEEE Transactions on Parallel and Distributed Systems, 17(8), 851-864, doi: 10.1109/TPDS.2006.103.

[192] Abdallah, W., Mnasri, S., & Val, T. (12/12/2020). Genetic-Voronoi algorithm for coverage of IoT data collection networks. In 2020 30th International Conference on Computer Theory and Applications (ICCTA) (pp. 16-22). IEEE, doi: 10.1109/ICCTA52020.2020.9477675.

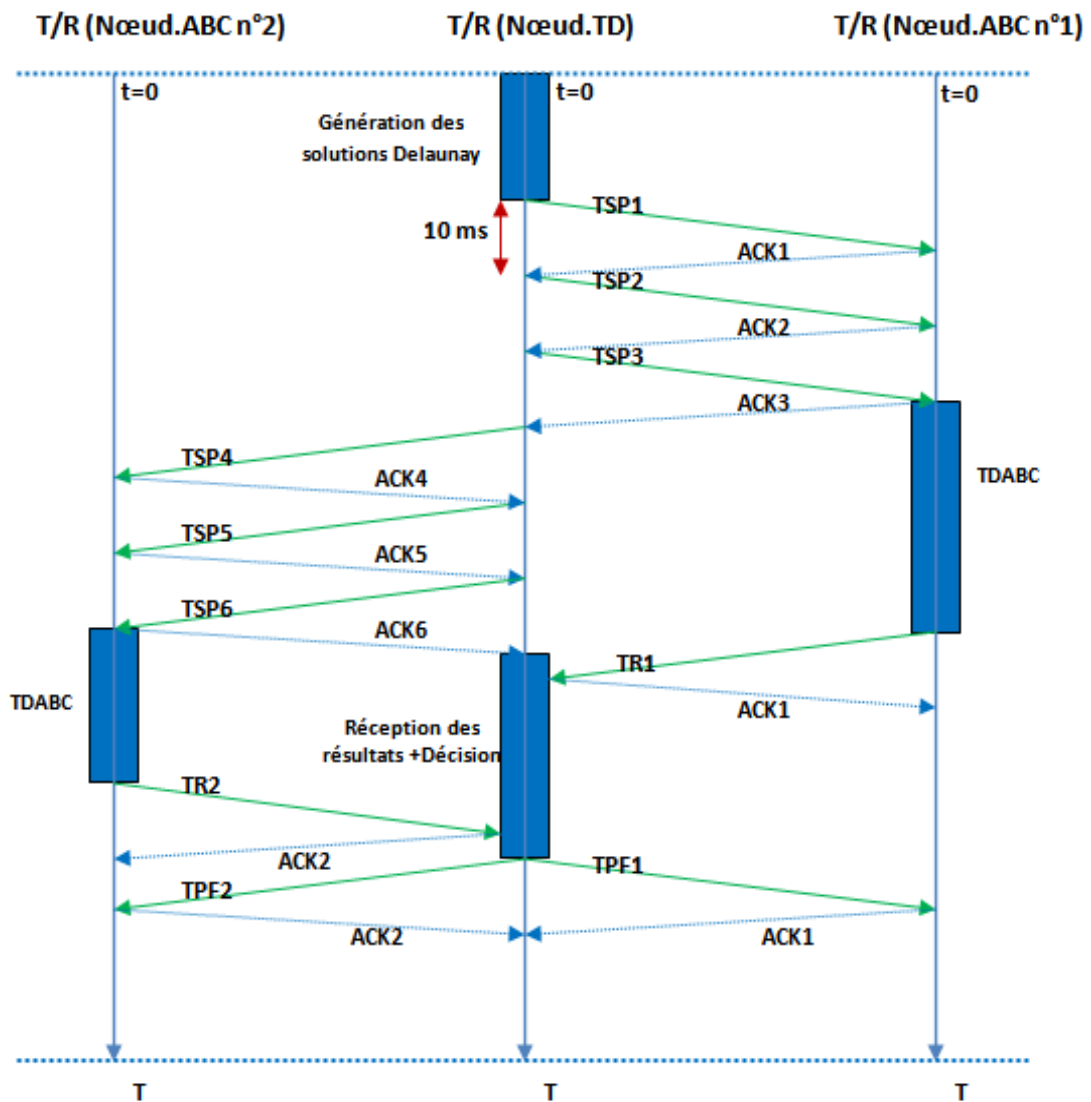
[193] Watson, D. F. (1981). Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. The computer journal, 24(2), 167-172, doi: 10.1093/comjnl/24.2.167

[194] Karaboga, D., & Ozturk, C. (2009). Neural networks training by artificial bee colony algorithm on pattern classification. Neural Network World, 19(3), 279.

- [195] Yu, X., Zhang, J., Fan, J., & Zhang, T. (2013). A faster convergence artificial bee colony algorithm in sensor deployment for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 9(10), 497264, doi: 10.1155/2013/497264
- [196] Simulateur OMNET ++ : <https://omnetpp.org/> (consulté le 20/05/2022)
- [197] Guerreiro, A. P., Fonseca, C. M., & Paquete, L. (2020). The hypervolume indicator: Problems and algorithms. arXivpreprint arXiv:2005.00515, doi: 10.1145/3453474
- [198] **Abdallah, W.**, Mnasri, S., & Val, T. (2022). Distributed approach for the indoor deployment of wireless connected objects by the hybridization of the Voronoi diagram and the Genetic Algorithm, *Journal of Engineering Research and Sciences*, Volume 1, Issue 2, Page # 10-23, 2022; doi: 10.55708/js0102002.
- [199] ESPNOW. Accessed : Mai 01, 2022 Available : <https://www.espressif.com/en/products/software/esp-now/overview>
- [200] M5StickC. (2020). Accessed: Mai 01, 2022. Available: <https://m5stack.com/products/stick-c>
- [201] PainlessMesh. Accessed: Mai 21, 2022. Available: <https://gitlab.com/painlessMesh/painlessMesh>
- [202] Fofana, N. I., Bossche, A. V. D., Dalcé, R., & Val, T. (2016, July). An original correction method for indoor ultra wide band ranging-based localisation system. In *International Conference on Ad-Hoc Networks and Wireless* (pp. 79-92). Springer, Cham, doi: 10.1007/978-3-319-40509-4\_6
- [203] Van den Bossche, A., Dalcé, R., Gonzalez, N., & Val, T. (2018, September). Locura: A new localisation and uwb-based ranging testbed for the internet of things. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (pp. 1-6). IEEE, doi: 10.1109/IPIN.2018.8533778.
- [204] Hassanat, A.; Almohammadi, K.; Alkafaween, E.; Abunawas, E.; Hammouri, A.; Prasath, V.B.S. (2019) Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach. *Information*, 10, 390. <https://doi.org/10.3390/info10120390>
- [205] Abu Alfeilat, H. A., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., & Prasath, V. B. S. (2019). Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. *Big Data*. doi:10.1089/big.2018.0175
- [206] Miller, B. L., & Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3), 193-212.
- [207] Blickle, T., & Thiele, L. (1996). A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4(4), 361-394.

# ANNEXES

Annexe 1 : Diagramme de séquence de protocole de communication entre trois nœuds pour l'hybridation TDABC







---

## Résumé

---

Les réseaux de collecte de données sont devenus récemment l'un des domaines de recherche importants grâce à leur rôle fondamental et à leurs larges applications dans de nombreux domaines. La mise en place de réseaux d'objets repose essentiellement sur le déploiement d'objets connectés pour traiter les données collectées et les transmettre à des différents équipements. Le déploiement 3D à faible coût pose de nombreux problèmes d'optimisation avec une littérature limitée. Notre objectif est donc de proposer une architecture la plus optimisée afin de fournir des solutions de capteurs en environnement 3D, tout en améliorant la performance du réseau. À cet égard, plusieurs questions et objectifs sont liés au problème du déploiement de réseaux de capteurs et sont traités dans les différents chapitres du manuscrit.

Ce dernier présente deux hybridations pour maximiser la couverture d'une région d'intérêt dans un réseau de collecte de données avec l'utilisation du minimum des nœuds. Une première hybridation nommée Diagramme Voronoi Algorithme Génétique (DVAG), entre une méthode géométrique de déploiement présentée par le diagramme de Voronoi et un algorithme d'optimisation présenté par l'algorithme génétique, est proposée dans une première approche centralisée puis une seconde distribuée. Une deuxième hybridation nommée Triangulation Delaunay Artificiel Bee Colonie (TDABC) combine une méthode géométrique de déploiement présentée par le diagramme de Triangulation de Delaunay avec un algorithme d'optimisation présenté par l'algorithme de colonie des abeilles artificielles ABC. Elle est introduite dans deux approches : centralisée puis distribuée. Dans l'approche centralisée, l'exécution se fait dans un seul nœud tandis que dans l'approche distribuée, l'exécution se fait dans plusieurs nœuds en parallèle. L'idée sur l'utilisation d'une approche distribuée pour le déploiement dans un réseau de collecte de données est de répartir/partager les tâches aux différents nœuds utilisés dans un réseau afin de gagner du temps et de l'énergie.

Les hybridations DVAG et TDABC développées ont été évaluées dans un environnement expérimental par prototypage sur un banc d'essai réel utilisant des nœuds M5StickC équipés d'un processeur ESP32. Les expérimentations montrent que l'approche distribuée fournit de meilleurs résultats qui sont donnés par les algorithmes originaux en termes de degré de couverture, de l'indicateur de puissance du signal reçu (RSSI), de durée de vie et de nombre d'objets voisins pour les deux hybridations. L'algorithme distribué de DVAG (D-DVAG) proposé surpasse celui de l'algorithme distribué de TDABC (D-TDABC) en termes de vitesse de calcul.

**Mots-clés:** algorithme d'optimisation, algorithmes génétiques, algorithme des abeilles artificielles, approche centralisée, approche distribuée, couverture, déploiement 3D à l'intérieur, diagramme de Voronoi, hybridation, méta-heuristique, réseaux de collecte de données, triangulation de Delaunay.

---

## Abstract

---

Recently, data collection IoT networks become one of the most relevant research fields due to their fundamental role and wide application. The management of connected objects in the networks is based on the deployment process to collect data and transmit them to different positions in the network. Low-cost 3D deployment poses many optimization problems with limited literature. Therefore, our goal is to find the most optimized architecture and provide 3D sensor deployment solutions, while improving the network performance. In this regard, several issues and objectives are considered in relation to the problem of deploying sensor networks.

This thesis introduces two hybridizations to maximize the coverage of a region of interest in a data collection IoT networks with a minimum number of objects. A first hybridization, named Voronoi Diagram Genetic Algorithm (DVAG), involves a geometric method of deployment, the Voronoi diagram, and an optimization algorithm, the genetic algorithm, with two versions, a centralized and a distributed one. A second hybridization, named Delaunay Triangulation Artificial Bee Colony (DTABC), combines another class of geometric deployment method, the Delaunay Triangulation diagram, with another optimization algorithm, the artificial bee colony algorithm. The latter hybridization is also implemented in a centralized and distributed version. In the centralized approach, the execution is achieved in a single node, while numerous nodes in a parallel execution are needed for the distributed approach. The distributed approach for the deployment in a data collection network allows sharing the tasks among the different nodes to enhance the execution time and the computation complexity.

The developed DVAG and TDABC hybridizations were evaluated in an experimental environment by prototyping on a real test bench using M5StickC nodes equipped with an ESP32 processors. The experiments show that the distributed approach provides better results than the original algorithms in terms of degree of coverage, Received Signal Strength Indication (RSSI), lifetime and number of neighboring objects. The proposed Distributed algorithm DVAG (D-DVAG) outperforms that of Distributed algorithm TDABC (D-TDABC) in terms of computational speed.

**Keywords :** artificial bee algorithm, centralized approach, coverage, data collection networks, Delaunay triangulation, distributed approach, genetic algorithms, hybridization, indoor 3D deployment, optimization algorithm, Voronoi diagram, meta-heuristics.