

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Electronique

Thèse de Doctorat en électronique

Les architectures parallèles embarquées dédiées à l'accélération des algorithmes évolutionnaires : Application à la conception d'un système dédié à l'agriculture de précision

Soumaya Ferhat-Taleb

Présenté et soutenu publiquement le (07/07/2025)

Composition du jury :

M. Cherif Larbes	Professeur	ENP	Président
Mme. Nour El-Houda Benalia	Docteur	ENP	Rapporteur
M. Mourad Adnane	Professeur	ENP	Examineur
Mme.Razika Boushaki	Professeur	UMBB	Examinatrice
Mme.Leila Ghoumari	MCA	ENSTA	Examinatrice
Mme.Imene Djelloul	MCA	ENSTA	Examinatrice
Mme.Souraya Hamida	MCA	UMK	Invitée
Mme.Nabila Naouar	MCB	ENP	Invitée

ENP 2025

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Electronique

Thèse de Doctorat en électronique

Les architectures parallèles embarquées dédiées à l'accélération des algorithmes évolutionnaires : Application à la conception d'un système dédié à l'agriculture de précision

Soumaya Ferhat-Taleb

Présenté et soutenu publiquement le (07/07/2025)

Composition du jury :

M. Cherif Larbes	Professeur	ENP	Président
Mme. Nour El-Houda Benalia	Docteur	ENP	Rapporteur
M. Mourad Adnane	Professeur	ENP	Examineur
Mme Razika Boushaki	Professeur	UMBB	Examinatrice
Mme Leila Ghoumari	MCA	ENSTA	Examinatrice
Mme Imene Djelloul	MCA	ENSTA	Examinatrice
Mme Souraya Hamida	MCA	UMK	Invitée
Mme Nabila Naouar	MCB	ENP	Invitée

ENP 2025

Dédicaces

*À mes parents,
Pour leur amour, leur patience et leurs prières.*

*À ma famille,
Pour leur soutien inestimable,
À mes deux anges,
Sydra et Yaman.*

Remerciements

La réalisation de cette thèse a été possible grâce à l'accompagnement, le soutien et l'encouragement de nombreuses personnes, à qui je souhaite exprimer ma profonde gratitude.

Je tiens d'abord à remercier mon directeur de thèse, **Dr. Nour el houda BENALIA**, pour sa guidance, ses conseils avisés et sa confiance tout au long de ces années. Ses compétences scientifiques et son soutien constant ont été essentiels à l'aboutissement de ce travail.

Je remercie également les membres du jury, pour l'honneur qu'ils m'ont fait d'accepter d'examiner ce travail. Leurs suggestions et commentaires constructifs ont grandement contribué à l'amélioration de cette thèse.

Un grand merci à tous mes collègues et amis du laboratoire, pour les discussions enrichissantes, le partage d'idées et l'ambiance de travail si précieuse. Leur soutien a rendu ces années de recherche plus agréables et motivantes.

Je souhaite exprimer ma reconnaissance à ma famille, en particulier à mes parents, pour leur amour inconditionnel, leur encouragement et leur confiance inébranlable. Sans leur soutien, je n'aurais pas pu mener à bien cette aventure.

Enfin, je remercie mes amis, proches et toutes les personnes qui, de près ou de loin, m'ont soutenu et encouragé tout au long de ce parcours.

À tous, merci du fond du cœur.

تواجه الزراعة الحديثة تحديات كبيرة نتيجة النمو السكاني السريع ونقص الموارد الطبيعية، خصوصاً المياه. لذلك، ظهرت الزراعة الذكية كحل واعد يعتمد على إدماج تقنيات حديثة لتحسين الإنتاج وضمان الاستدامة. من بين هذه التقنيات، تلعب الشبكات اللاسلكية للمستشعرات (WSN) دوراً حاسماً في مراقبة التربة. في هذا العمل، قمنا باقتراح أداة لمساعدة المصممين على تحسين عملية نشر العقد في شبكة WSN مخصصة لتطبيق زراعي لمراقبة التربة.

في الجزء الأول، اقترحنا نموذجاً رياضياً يأخذ بعين الاعتبار عدة قيود تشمل التكلفة، التغطية، الاتصال، التغطية الزائدة، بالإضافة إلى قيد جديد يتعلق بالمسافة بين العقد. في الجزء الثاني، استُخدمت خوارزميات تطورية متعددة الأهداف (MOEAs) مثل NSGA-II و SPEA-II، وقد أظهرت أداءً جيداً. ولكن لوحظ وجود بطء في التنفيذ عند معالجة مساحات زراعية واسعة. لهذا السبب، في الجزء الثالث، تم اعتماد تسريع برمجي عبر تهجين SPEA-II مع خوارزميات تعلم آلي للتقريب (MLP)، (KNN)، (DT)، (RF)، مما أدى إلى تقليص زمن التنفيذ بنسبة 66%. كما تم تنفيذ تسريع مادي باستخدام معالجات GPU لتحقيق تحسن إضافي بنسبة 28%.

الكلمات المفتاحية: الزراعة الذكية، الشبكات اللاسلكية، خوارزميات تطورية، تحسين متعدد الأهداف، التعلم الآلي، تسريع، نشر العقد.

Abstract:

Modern agriculture faces major challenges due to rapid population growth and the depletion of natural resources, especially water. Smart agriculture has emerged as a promising solution by integrating advanced technologies to improve yields and ensure sustainability. Among these technologies, Wireless Sensor Networks (WSNs) are essential for soil monitoring. In this thesis, we proposed a tool to support designers in optimizing node deployment in a WSN for an agricultural soil monitoring application.

In the first part, we introduced a mathematical model that considers several constraints, including cost, coverage, connectivity, over-coverage, and a newly introduced constraint: the distance between nodes. In the second part, we used Multi-Objective Evolutionary Algorithms (MOEAs) such as NSGA-II and SPEA-II, which demonstrated good performance. However, execution latency was observed for large agricultural areas. Therefore, in the third part, we proposed software acceleration through the hybridization of SPEA-II with machine learning regression algorithms (MLP, KNN, DT, RF), achieving a 66% reduction in execution time. Hardware acceleration using GPUs provided an additional 28% speed-up.

Keywords: Precision Agriculture, Wireless Sensor Networks, Evolutionary Algorithms, Multi-objective Optimization, Machine Learning, Acceleration, Node Deployment.

Résumé :

L'agriculture moderne est confrontée à de nombreux défis liés à la croissance rapide de la population et à la pénurie des ressources naturelles, notamment l'eau. L'agriculture de précision s'impose alors comme une solution prometteuse intégrant des technologies avancées pour améliorer les rendements et préserver les ressources. Parmi ces technologies, les réseaux de capteurs sans fil (WSN) jouent un rôle essentiel dans la surveillance des sols. Dans cette thèse, nous avons proposé un outil d'aide au déploiement optimal des nœuds dans un WSN destiné à une application agricole de surveillance du sol.

Dans la première partie, un modèle mathématique a été développé en intégrant plusieurs contraintes : coût, couverture, connectivité, sur-couverture, et une nouvelle contrainte de distance entre les nœuds. Dans la deuxième partie, nous avons appliqué des algorithmes évolutionnaires multi-objectifs (MOEAs) tels que NSGA-II et SPEA-II, qui ont montré de bonnes performances, bien que ralentis pour de grandes surfaces. Ainsi, dans la troisième partie, nous avons proposé une hybridation logicielle du SPEA-II avec des algorithmes de régression issus de l'apprentissage automatique (MLP, KNN, DT, RF), ce qui a permis une réduction de 66 % du temps d'exécution. Une accélération matérielle via GPU a permis un gain supplémentaire de 28 %.

Mots-clés : Agriculture de précision, réseaux de capteurs sans fil, algorithmes évolutionnaires, optimisation multi-objectifs, apprentissage automatique, accélération, déploiement de nœuds.

Publications et communications durant la thèse

Articles

- Taleb, S. F., Benalia, N. E. H., Sadoun, R. (2023). Fitness approximation with RF algorithm dedicated to WSN node deployment for a soil monitoring application. *International Journal of Ad Hoc and Ubiquitous Computing*, 43(2), 72-86.
- Ferhat Taleb, Soumaya, Nour El-Houda Benalia, and Rabah Sadoun. "Evolutionary algorithm applications for IoTs dedicated to precise irrigation systems : state of the art." *Evolutionary Intelligence* 16.2 (2023) : 383-400.
- Taleb, S. F., Benalia, N. (2023). Wireless sensor network deployment optimization for a smart farming application : comparison of two Multi-Objective Evolutionary Algorithms. *ENP Engineering Science Journal*, 3(2), 15-24.
- BENALIA, Nour El-Houda, TALEB, Soumaya Ferhat, TIBHIRT, Amel, et al. Diabetes Prediction Tool under System on Chip Using Machine Learning Method. *Artificial Intelligence Evolution*, 2022, p. 69-86.
- BENALIA, Nour El-Houda, SI HADJ MOHAND, Imene, FERHATTALEB, Soumaya, et al. MoEA-DeployWSN-SB : Three variants of multi-objective evolutionary algorithms for the deployment optimization strategy of a WSN in a smart building. *International Journal of Information Technology*, 2022, vol. 14, no 1, p. 333-344.

Conférences internationales

- International seminar on mathematics and informatics 2021, SOUMAYA FERHAT TALEB, NOUR EL-HOUDA BENALIA, RABAH SADOON, Optimization algorithm applications for smart irrigation, 340-346.
- International seminar on mathematics and informatics 2021, SOUMAYA FERHAT TALEB, NOUR EL-HOUDA BENALIA, RABAH SADOON, Défis de déploiement des réseaux de capteurs sans fils dédiée à l'agriculture de précision.(Poster)
- international seminar on mathematics and informatics 2021, Nour El-Houda BENALIA, Soumaya FERHAT-TALEB, Imene SI HADJ MOHAND, Rabah SADOON, Effective deployment strategy for WSNs based on multi-objective Eas in smart building, 72-85.

Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

Introduction générale		18
1 Agriculture de précision et déploiement des WSN		22
1.1	Introduction	23
1.2	Agriculture de précision	23
1.2.1	Domaines d'application de la PA	25
1.2.1.1	Gestion des cultures	26
1.2.1.2	Contrôle de l'irrigation	27
1.2.1.3	Surveillance du sol	29
1.2.1.4	Détection des maladies	31
1.2.1.5	Suivie du bétail	32
1.2.2	Avantages de la PA	34
1.2.3	Outils et technologies pour la PA	35
1.2.3.1	Outils d'intervention dans les champs (les robots agri- coles)	36
1.2.3.2	Outils de collecte des données	38
1.3	Architecture de base d'un WSN	46
1.4	Éléments constructifs des WSNs	48
1.4.1	Modes d'installation des WSN	48
1.4.1.1	Réseaux de capteurs sans fil terrestres (TWSN)	49
1.4.1.2	Réseaux de capteurs sans fil souterrain (UWSN)	49
1.4.2	Architecture de déploiement des WSNs	50
1.4.2.1	Architecture stationnaire	50
1.4.2.2	Architecture mobile	51
1.4.2.3	Architecture hybride	51
1.4.3	Technologies et normes de communication des nœuds dans les WSNs	51
1.4.3.1	Zigbee-IEEE 802.15.4	51
1.4.3.2	Wifi	52
1.4.3.3	Bluetooth	52

1.4.3.4	GPRs \3g\4g	53
1.4.3.5	WiMAX	53
1.4.4	Topologie des WSN	55
1.4.4.1	Étoile (Star)	55
1.4.4.2	Arbre (Tree)	55
1.4.4.3	Maille (Mesh)	55
1.4.4.4	Topologie hybride	56
1.4.5	Alimentation des WSNs	56
1.4.6	Défis des WSNs	57
1.4.7	Majeurs objectifs de déploiement des WSNs	59
1.4.7.1	Couverture	59
1.4.7.2	Connectivité	62
1.4.7.3	Durée de vie	63
1.4.8	Domaines d'utilisation des WSN	63
1.4.8.1	Applications militaire et prévention de la criminalité	64
1.4.8.2	Applications en réseau de zones corporelles	65
1.4.8.3	Applications en agriculture	65
1.4.8.4	Application dans l'environnement	67
1.4.8.5	Applications en urbanisation et infrastructure	68
1.5	Techniques de déploiement des WSNs	69
1.5.1	Déploiement statique avec des méthodes exactes	71
1.5.1.1	Méthode de gradient	71
1.5.1.2	Méthode du simplexe	71
1.5.1.3	Branch et Bound	72
1.5.1.4	Méthode de diagramme de Voronoi	73
1.5.2	Déploiement statique avec des méthodes heuristiques	73
1.5.2.1	Recherche Tabou	74
1.5.2.2	Intelligence en Essaim	75
1.5.2.3	Algorithmes d'optimisation multi-objectifs	75
1.5.3	Déploiement statique aléatoire	76
1.6	Conclusion	76

2 Application des MOEAs pour l'optimisation du déploiement des nœuds 78

2.1	introduction	79
2.2	Optimisation des problèmes multi-objectifs	79
2.2.1	Paramètres de l'optimisation multi-objective	81
2.2.1.1	Dominance au sens de Pareto	81
2.2.1.2	Pareto optimal	82
2.2.1.3	Front de Pareto	82
2.3	Méthodes de résolutions des MOPs	83
2.3.1	Méthodes déterministe	83
2.3.2	Méthodes méta-heuristiques	84
2.4	Algorithmes évolutionnaires (Evolutionary Algorithms, EAs)	85
2.4.1	Composants de base d'un EA	87
2.4.1.1	Chromosome	87
2.4.1.2	Fonctions objectives	88

2.4.1.3	Opérateurs génétiques	88
2.4.1.4	Critères d'arrêt	89
2.4.2	Principe de fonctionnement d'un EA	89
2.5	Algorithmes évolutionnaires multi-objectifs (Multi-objectif Evolutionary Algorithms, MOEAs)	91
2.5.1	MOEAs basés sur la décomposition	92
2.5.2	MOEAs basés sur la dominance	92
2.6	Problème d'optimisation de déploiement d'un WSN pour l'agriculture	94
2.6.1	Algorithmes utilisés	95
2.6.1.1	Strength Pareto Evolutionary Algorithm II (SPEA II)	95
2.6.1.2	Non Sorting Genetic Algorithm II (NSGA II)	97
2.6.2	Étude des travaux antérieurs	98
2.7	Adaptation des algorithmes MOEAs au problème de déploiement	102
2.7.1	Modélisation de l'espace de déploiement	102
2.7.2	Initialisation de la population	103
2.7.3	Modélisation des fonctions objectives	104
2.7.3.1	Nombre de nœuds de capteur	105
2.7.3.2	Distance de séparation entre les capteurs	105
2.7.3.3	Couverture et sur-couverture	106
2.7.3.4	Taux de communication des nœuds	107
2.7.4	Opérateurs génétiques	108
2.7.4.1	Sélection des parents	108
2.7.4.2	Opérateur de croisement	109
2.7.4.3	Opérateur de mutation	110
2.7.5	Critères d'arrêt	110
2.8	Comparaison entre SPEA II et NSGA II	111
2.8.1	Définition des hyper-paramètres MOEAs	112
2.8.2	Résultats et discussion	113
2.8.3	Limite du modèle proposé	117
2.9	Conclusion	119
3	Accélération des MOEAs	121
3.1	Introduction	122
3.2	Hybridation des méta-heuristiques	124
3.3	Problèmes de conception	125
3.3.1	Classification hiérarchique des méta-heuristiques	125
3.3.1.1	Hybridation relais de bas niveau	126
3.3.1.2	Hybridation co-évolutionnaire de bas niveau	126
3.3.1.3	Hybridation relais de haut niveau (High-level Relay Hybrid)	127
3.3.1.4	Hybridation co-évolutionnaire de haut niveau (High-level Teamwork Hybrid)	128
3.3.2	Classification à plat des méta-heuristiques	128
3.4	Approximation de la valeur d'évaluation	130
3.4.1	Méthodes d'approximation	132
3.4.1.1	Modèles analytiques	132
3.4.1.2	Échantillonnage Aléatoire	132

3.4.1.3	Méthodes d'Extrapolation	132
3.4.1.4	Apprentissage Machine	133
3.4.2	Types d'approximation	133
3.4.2.1	Approximation du problème	133
3.4.2.2	Approximation fonctionnelle	133
3.4.2.3	Approximation évolutive	134
3.4.3	Incorporation de modèles de fitness approximatifs	135
3.4.3.1	Contrôle d'évolution	136
3.4.4	Apprentissage automatique supervisé	137
3.4.5	Approximation des valeurs de fitness par SML	137
3.4.5.1	Régression linéaire	138
3.4.5.2	Réseaux de neurones (ANN)	138
3.4.5.3	k-plus proches voisins (KNN)	138
3.4.5.4	Arbres de décisions (DT)	139
3.4.5.5	Forêt aléatoire (RF)	141
3.5	Solution logicielle proposée	142
3.5.1	Etude de l'existant	142
3.5.2	Description de la solution	146
3.5.3	Définition du taux de contrôle de l'évolution	148
3.5.4	Déroulement des Différentes Phases de SPEA II hybride avec Contrôle d'Évaluation	149
3.5.4.1	Génération de la Population Initiale	150
3.5.4.2	Formation Initiale des modèle de régression	150
3.5.4.3	Phase Évolutive	150
3.5.4.4	Évaluation avec Contrôle d'Évolution	150
3.5.4.5	Ré-entraînement Continu du modèle d'approximation	150
3.5.4.6	Sélection	151
3.5.4.7	Boucle Itérative	151
3.6	Implémentation de la solution logicielle	152
3.6.1	Cas d'étude	152
3.6.1.1	Définition des Hyper-paramètres	153
3.6.2	Interprétation des résultats	154
3.6.3	Comparaison entre les différents algorithmes de régression	158
3.6.3.1	Comparaison en utilisant la métrique de R-squared	158
3.6.3.2	Comparaison en utilisant la métrique de l'erreur moyenne quadratique	160
3.6.4	Amélioration des résultats obtenues par RF-SPEA II	162
3.6.4.1	Sélection de caractéristiques	163
3.6.4.2	Ajustement des Hyper-paramètres	165
3.7	Modèle de parallélisations des MOEAs	170
3.7.1	Origine et parallélisme des EAs	170
3.7.2	Classification des modèles d'EA parallèles	172
3.7.2.1	Modèle Maître-esclave	173
3.7.2.2	Modèle en îlots	174
3.7.2.3	Modèle totalement distribué	174
3.7.3	Relation entre le modèle EA parallèle et l'architecture parallèle ciblée	175

3.7.4	Architectures sous-jacentes existantes : un aperçu	175
3.8	Calcule distribué	176
3.8.1	Eléments de parallélisme	177
3.8.1.1	Mémoires des architecteurs parallèles	177
3.8.1.2	Implémentations d'EA parallèles	178
3.9	Graphic Processing Unit (GPU)	180
3.9.1	Architecture de base du GPU	181
3.10	Parallélisations du modèle de déploiement	182
3.10.1	Étude de l'existant	182
3.10.2	Architecture parallèle proposée	183
3.10.2.1	Temps d'exécution	184
3.10.2.2	Nombre de capteurs	185
3.10.2.3	Couverture	185
3.10.2.4	Sur-couverture	185
3.10.2.5	Connectivité	186
3.10.2.6	Distance	186
3.11	Conclusion	187
Conclusion générale et perspectives		189
Bibliographie		192

Liste des tableaux

1.1	Exemples des travaux de la littérature qui ont bénéficié des compétences des robots agricoles dans l'agriculture.	39
1.2	Exemples des travaux de la littérature qui ont bénéficié des compétences de la technologie de la télédétection dans l'agriculture.	43
1.3	Exemples des travaux de la littérature qui ont bénéficié des compétences de la technologie WSN dans la PA.	47
1.4	Comparaison entre les modes d'installation des WSN.	50
1.5	Technologies et normes de communication utilisées dans les WSNs pour une application agricole.	54
1.6	Comparaison entre les batteries rechargeables utilisées dans les WSN. .	57
2.1	Applications des GAs pour l'optimisation du déploiement des WSNs. .	100
2.2	Comparaison entre SPEA II et NSGA II en fonction de la qualité des solutions et du temps d'exécution (en secondes).	116
2.3	La relation entre l'augmentation de la taille des parcelles, le temps d'exécution et la qualité des solutions pour les mêmes hyper-paramètres.	118
3.1	Répartition des travaux selon le type de variante des GAs, la technique d'approximation et le domaine d'application.	145
3.2	Impact du taux de contrôle sur la qualité de la meilleure solution et le temps d'exécution.	148
3.3	Comparaison entre SPEA II et SPEA II hybride avec les différents algorithmes de régression pour une taille de parcelle de 400 m^2	154
3.4	Comparaison entre RF-SPEA II et RF-SPEA II (GiNI) pour une taille de parcelle de 400 m^2	164
3.5	Comparaison entre RF-SPEA II (a) et RF-SPEA II (b) avec des Hyper-paramètres différents pour une taille de parcelle de 400 m^2	166
3.6	Comparaison entre l'exécution du SPEA II sur deux types de matériel : CPU et GPU. pour une taille de parcelle de 400 m^2	185

Table des figures

1.1	Techniques de la PA.	26
1.2	Appareil de calcul de LAI.	27
1.3	Contrôle de l'irrigation par tuyaux.	28
1.4	Système de surveillance de l'humidité du sol.	30
1.5	Détection des maladies en utilisant des images thermique.	32
1.6	Surveillance en élevage.	33
1.7	Répartition des technologies utilisées dans la PA.	36
1.8	Plantation des grains avec un robot agricole	37
1.9	Exemple d'un système de télédétection satellitaire.	40
1.10	Exemple d'un TWSN appliqué à la PA.	44
1.11	Modèle architectural d'un WSN.	46
1.12	Architecture de base d'un nœud capteur.	48
1.13	Défis du WSN.	58
1.14	Modèle de couverture de champ.	60
1.15	Modèle de couverture des points d'intérêts (cibles).	61
1.16	Modèle de couverture de barrières.	61
1.17	Taxonomie des applications des WSNs.	64
1.18	Surveillance militaire par WSN	64
1.19	Télémédecine par WSN.	65
1.20	Systèmes de surveillance du sol par WSN.	67
1.21	Application du WSN pour la prédiction de la météo.	68
1.22	Façade du bâtiment The Edge. Photo de Ronald Tilleman.	69
1.23	Méthodes de déploiement des nœuds dans un WSN.	70
1.24	La division du problème P en plusieurs sous-problèmes par l'algorithme Branch Bound.	72
2.1	Exemple de front de Pareto [1]	82
2.2	Classification des méthodes d'optimisation.	83
2.3	Différentes familles des EAs.	86
2.4	Fonctionnement des EAs.	90
2.5	Principe de fonctionnement de l'algorithme SPEA-II.	96
2.6	Principe de fonctionnement de l'algorithme NSGA II.	98
2.7	Comparaison des types de GA pour le déploiement WSN.	99
2.8	Pourcentage d'utilisation des différentes contraintes dans les 16 ou- vrages étudiés.	101
2.9	Grille de déploiement des nœuds.	103

2.10	Formulation et codage des chromosomes.	104
2.11	Exemple d'un individu généré aléatoirement de la population initiale (rendement de la dimension 8 m ²).	104
2.12	Distance d'encombrement.	109
2.13	Opérateur de croisement et de mutation.	110
2.14	Front de Pareto d'un ensemble de solutions pour un exemple de taille de parcelle de 32m ²	111
2.15	Impact de la taille de la population sur la qualité de la meilleure solution et le temps d'exécution.	112
2.16	Comparaison entre NSGA II et SPEA II en fonction de l'évolution de leurs fonctions objectives.	114
2.17	Comparaison entre les solutions non dominées NSGA II et SPEA II selon leurs fonctions objectives.	116
2.18	Plan de déploiement du nœud capteur pour une parcelle agricole de 100 m ²	117
2.19	Approches pour réduire le temps d'exécution des MOEAs.	119
3.1	Schéma de classification hiérarchique des méta-heuristiques hybrides. .	126
3.2	Schéma de classification horizontale (plate) des méta-heuristiques hy- brides.	129
3.3	Faux Optimum d'un modèle approximatif.	136
3.4	Méthode d'agrégation Bootstrap.	140
3.5	Processus du Boosting de plusieurs modèles d'apprentissage.	141
3.6	Schéma de fonctionnement du modèle RF.	142
3.7	Impact du taux de contrôle sur la meilleure qualité de la solution et le temps d'exécution.	149
3.8	Architecture de la solution logicielle.	151
3.9	Pseudo-code du RF-SPEA II hybride	152
3.10	Comparaison entre l'évolution des fonction objectives de SPEA II et SPEA II hybride avec les différents algorithmes de régression pour une taille de parcelle de 400 m ²	157
3.11	Comparaison de la métrique de R-squared des différents algorithmes de régression pour une taille de parcelle de 400 m ²	160
3.12	Comparaison de la métrique de l'erreur moyenne quadratique des différents algorithmes de régression pour une taille de parcelle de 400 m ² .161	161
3.13	Importance des caractéristiques avec le modèle approximatif RF pour un exemple de taille de parcelle de 12 m ²	164
3.14	L'évolution des fonction objectives du RF-SPEA II (GINI) pour une parcelle agricole de taille de 400 m ²	165
3.15	Évolution des Performances de l'Algorithme RF-SPEA II (GINI) au Fil des Générations	165
3.16	L'évolution des fonctions objectives de RF-SPEA II pour différents hyper-paramètres.	166
3.17	Évolution du RMSE selon les générations pour une taille de parcelle de 400 m ²	168

3.18	Les formes de parallélisations des EAs.	173
3.19	Exemple d'architecture parallèle à mémoire partagée.	177
3.20	Exemple d'architecture parallèle à mémoire distribuée.	178
3.21	Architecture de base d'un GPU.	181
3.22	Pourcentage de l'utilisation des plates-formes matérielles de calcul distribué pour l'optimisation du temps d'exécution des MOEAs [2].	183
3.23	Architecture proposée de la solution matérielle.	184

Liste des abréviations

AI : Artificial Intelligence
ANN : Artificial Neural Network
APF : Artificial Potential Fields
CPU : Control Processing Unit
DT : Decision Tree
EAs : Evolutionary Algorithms
ET : Evapotranspiration
FPGA : Field Programmable Gate Array
GA : Genetic Algorithms
GNSS : Global Navigation Satellite System
GPS : Global positioning system
GPRS : General Packet Radio Service
GPU : Graphic Processing Unit
IoT : Internet of Things
ISM : Industrial, Scientific, and Medical
KNN : K-Nearest Neighbors
LAI : Leaf Area Index
MLP : Multilayer Perceptron
MOEAs : Multi-objectif Evolutionary Algorithms
MOGA : Multi-Objective Genetic Algorithms
NDVI : Normalized Difference Vegetation Index
NSGA II : Non-dominated Sorting Genetic Algorithm II
PA : Precise Agriculture
PI : Precise Irrigation

PLF : Precision Livestock Farming
PSO : Particle Swarm Optimization
Rd : Rayon de détection
Rc : Rayon de communication
RF : Random Forest
SAVI : Soil Adjusted Vegetation Index
SML : Supervised Machine Learning
SO : Stochastic Optimization
SPEA II : Strength Pareto Evolutionary Algorithm II
TWSN : Terrestrial Wireless Sensor Networks
UAS : Unmanned Aerial System
UAV : Unmanned Aerial Vehicle
UWSN : Underground Wireless Sensor Network
WiFi : Wireless Fidelity
Wimax : Worldwide Interoperability for Microwave Access
WSN : Wireless Sensor Network

Introduction générale

Introduction générale

Les besoins en production alimentaire augmentent avec la croissance démographique mondiale, qui devrait atteindre entre 9,4 et 10,2 milliards d'ici 2050 [1]. Pour répondre à ces défis, de nouvelles techniques agricoles ont été adoptées afin de résoudre les problèmes liés aux terres arables, à l'eau et au changement climatique, qui menacent les objectifs de l'agriculture durable. Parmi ces innovations, l'agriculture de précision (PA) s'est imposée comme une solution efficace. L'agriculture de précision vise à optimiser la production agricole en surveillant et en gérant précisément des facteurs comme l'irrigation, l'application d'engrais et de pesticides, ainsi que la croissance des cultures. Cette approche repose sur des mesures directes ou indirectes fournissant des informations en temps réel sur les cultures [2].

Pour mettre en œuvre l'agriculture de précision, diverses technologies sont employées, notamment les réseaux de capteurs sans fil (WSN). Ces technologies utilisent un ensemble de composants pour collecter, surveiller et analyser les données détectées [3]. Dans le domaine agricole, l'intégration de l'Internet des objets (IoT) avec les WSN permet d'optimiser la qualité des cultures et de préserver les ressources naturelles. Des capteurs surveillent des paramètres physiques vitaux tels que l'humidité, la température et le pH, pour déterminer les besoins précis des plantes.

Cependant, malgré ses avantages, la technologie WSN rencontre des défis importants, notamment le déploiement des nœuds dans la région d'intérêt (RoI). Ce déploiement est un problème NP-difficile qui affecte la connectivité, la couverture, la qualité de service et la durée de vie du réseau. Pour surmonter ces obstacles, des algorithmes d'optimisation multi-objectifs stochastiques, tels que les algorithmes évolutionnaires multi-objectifs (MOEA), sont utilisés. Les MOEA sont réputés pour

leur flexibilité et leur capacité à proposer plusieurs solutions optimales de Pareto, facilitant ainsi une recherche moins subjective dès les premières phases. Cependant, plus le problème est complexe, plus les MOEA nécessitent une grande puissance de calcul et un temps d'exécution important [4].

Pour surmonter ces limitations, notre recherche se concentre sur l'accélération des MOEAs via deux approches complémentaires : l'hybridation avec des algorithmes d'apprentissage automatique, en particulier des algorithmes de régression, et l'utilisation de GPUs pour le calcul parallèle. L'hybridation avec des algorithmes de régression vise à améliorer la convergence des MOEAs en réduisant le temps nécessaire pour atteindre des solutions de haute qualité, tandis que l'utilisation des GPUs permet de tirer parti des capacités de calcul massivement parallèle pour accélérer l'exécution des algorithmes.

Cette thèse propose une exploration approfondie de ces stratégies d'accélération, en les appliquant spécifiquement à des problèmes de déploiement de WSN dans le contexte de l'agriculture de précision. L'objectif est de démontrer que ces approches peuvent non seulement réduire significativement le temps d'exécution des MOEAs, mais aussi maintenir, voire améliorer, la qualité des solutions obtenues. En combinant des techniques avancées d'optimisation, d'apprentissage automatique, et de calcul haute performance, cette recherche contribue à rendre les MOEAs plus praticables pour des applications réelles à grande échelle, ouvrant ainsi la voie à une agriculture de précision plus efficace et durable.

Problématique

Cette recherche aborde deux problématiques principales. La première concerne l'optimisation du déploiement des nœuds pour la surveillance des sols en agriculture de précision, en utilisant des variantes d'algorithmes évolutionnaires multi-objectifs tout en respectant diverses contraintes liées à l'application. La deuxième problématique consiste à proposer des solutions pour accélérer l'exécution des MOEA appliqués à notre modèle de déploiement.

Objectifs

- 1. Modélisation des WSNs :** Mettre en œuvre une modélisation des WSNs et de l'environnement de l'application en tenant compte des contraintes de déploiement des nœuds telles que la couverture, la sur-couverture, la connectivité, le coût du réseau et la distance entre les nœuds capteurs, impactant la qualité des modèles de prévision.
- 2. Optimisation et accélération :** Proposer des solutions matérielles et logicielles pour accélérer les MOEA, en particulier pour les grandes parcelles agricoles, et valider ces solutions par une étude comparative.

Contributions de la thèse

- 1. Modèle de déploiement des nœuds :** Conception d'un modèle intégrant une nouvelle contrainte de déploiement, la distance de séparation des nœuds.
- 2. Application des MOEA :** Optimisation du déploiement des nœuds avec le Non-dominated Sorting Genetic Algorithm II (NSGA II) et le Strength Pareto Evolutionary Algorithm II (SPEA II) selon cinq contraintes [1].
- 3. Hybridation SPEA II :** Accélération logicielle du modèle avec l'hybridation de SPEA II avec quatre algorithmes de regression comme le Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Decision Tree (DT) et le Random Forest Regressor (RF) [2].
- 4. Architecture parallèle :** Proposition d'une architecture parallèle pour le modèle sur des plateformes de calcul distribué, permettant une accélération matérielle.

Organisation du manuscrit

- **Chapitre 1 :** Définition du contexte de recherche et des problématiques du déploiement des capteurs sans fil en agriculture de précision, suivi des techniques existantes pour le déploiement des nœuds.
- **Chapitre 2 :** Étude des travaux sur les algorithmes évolutionnaires pour l'optimisation du déploiement des nœuds, modélisation et développement de notre outil, comparaison entre NSGA II et SPEA II selon la qualité des plans de déploiement et leur temps d'exécution, et limites de notre modèle.
- **Chapitre 3 :** Ce chapitre est divisé en deux parties, la première présente une solution logicielle pour résoudre la lenteur d'exécution, avec une architecture hybride

utilisant quatre algorithmes de régression et une étude comparative. La deuxième partie présente une solution matérielle pour l'accélération du modèle, étude comparative des travaux sur le parallélisme des EAs, proposition d'une architecture parallèle et comparaison des résultats des deux méthodes d'accélération pour un cas d'étude.

● **Conclusion générale :** La thèse se termine par une conclusion générale qui synthétise les résultats obtenus tout au long du travail de recherche. Elle met en avant les principales contributions, les gains en performance réalisés, et ouvre des perspectives pour des travaux futurs visant à améliorer davantage le processus de déploiement et son intégration dans des systèmes réels.

Chapitre 1

Agriculture de précision et déploiement des WSN

1.1 Introduction

L'agriculture de précision (Precise Agriculture, PA) représente une révolution dans le domaine de l'agriculture, intégrant des technologies avancées pour optimiser les pratiques agricoles, augmenter les rendements et réduire l'impact environnemental. Cette approche repose sur la collecte et l'analyse de données précises pour prendre des décisions informées, en temps réel, concernant chaque aspect de la production agricole. Parmi les technologies de pointe employées, les réseaux de capteurs sans fil (Wireless Sensor Network, WSN) occupent une place centrale. Les WSNs permettent de surveiller en continu des paramètres cruciaux tels que l'humidité du sol, la température, la luminosité et la détection de maladies. Ces capteurs, répartis stratégiquement dans les champs, envoient des données à un système central, facilitant ainsi une gestion agricole plus précise et réactive. Cependant, la mise en place efficace de ces réseaux pose des défis importants. Le déploiement optimal des capteurs est essentiel pour garantir une couverture complète et une collecte de données fiable. Un déploiement aléatoire peut conduire à des zones non couvertes et une inefficacité énergétique, tandis qu'un déploiement déterministe, bien que plus précis, peut être difficile à réaliser, surtout dans des environnements inaccessibles ou sensibles. C'est ici qu'interviennent les techniques d'optimisation avancées. Ce chapitre explore les différentes technologies utilisées dans la PA, en mettant un accent particulier sur les WSNs et les défis liés à leur déploiement. Nous examinons également les méthodes d'optimisation, telles que les algorithmes d'optimisation multi-objectifs, qui offrent des solutions robustes pour le placement stratégique des capteurs. En comprenant et en maîtrisant ces techniques, il est possible de maximiser l'efficacité des WSNs, contribuant ainsi à une PA plus durable et productive.

1.2 Agriculture de précision

Les besoins en production alimentaire augmentent avec la croissance de la population, qui devrait atteindre entre 9,4 et 10,2 milliards d'ici 2050 [3]. En conséquence, de nouvelles techniques agricoles ont été adoptées pour résoudre les problèmes liés à la limitation des terres, au manque de ressources en eau et au changement climatique,

des enjeux qui représentent un danger considérable pour l’accomplissement des objectifs d’une agriculture durable. Aux États-Unis, la PA est née dans les années 1980. En 1985, des chercheurs de l’Université du Minnesota ont commencé à varier les apports d’amendements calciques sur des parcelles agricoles [4].

La PA vise à optimiser les rendements en surveillant les différentes phases de la production agricole grâce à la mesure directe ou indirecte de plusieurs variables fournissant des informations immédiates sur l’état et les besoins exacts des cultures [5]. Pour optimiser ce processus, il a été nécessaire d’adopter de nouvelles technologies capables de faciliter et d’optimiser des tâches précises telles que la plantation automatique à l’aide de systèmes mondiaux de navigation par satellite (Global Navigation Satellite System, GNSS), la collecte des cultures à l’aide de capteurs embarqués sur des machines agricoles, la détection des maladies par imagerie satellitaire et drones agricoles [6], et la surveillance des sols via des WSNs [7].

La plupart de ces technologies génèrent une grande quantité de données essentielles pour l’application de la PA. Il est donc crucial d’utiliser des techniques de traitement des données émergentes telles que l’analyse des données massives (Big Data), le Cloud Computing, l’intelligence artificielle (Artificial Intelligence, AI) et les algorithmes d’apprentissage automatique, qui permettent de tirer des informations utiles pour faciliter la production [8, 9].

La PA se définit comme la gestion optimale des parcelles agricoles pour maximiser les rendements tout en préservant les ressources naturelles, énergétiques et financières, en tenant compte de la variabilité entre les parcelles et des différences de croissance des plantes dans un même champ [10]. Elle vise à optimiser l’utilisation des ressources disponibles pour accroître la rentabilité et la durabilité des exploitations agricoles, tout en réduisant les impacts environnementaux négatifs et en améliorant la qualité des sols ainsi que les aspects sociaux de l’agriculture, de l’élevage et des professions associées [11]. De plus, les nouvelles technologies appliquées en PA ont apporté des solutions spécifiques à divers problèmes du processus agricole, comme la surveillance des sols [12], la détection des maladies [13], l’application d’engrais [14] et la PI [15].

1.2.1 Domaines d'application de la PA

La croissance de la demande des produits alimentaires exerce une énorme pression sur les producteurs agricoles, spécialement avec l'augmentation de la taille de la population, la réduction de la superficie des terres arables et la pénurie de certaines ressources, ce qui les a incités à augmenter la production agricole en utilisant les engrais et autres produits chimiques de manière aléatoire. Par conséquence, cette sur-utilisation qui ne respecte pas forcément la durabilité écologique a un effet négatif sur la santé des sols et par conséquence la santé des plantes [16].

La nature dynamique de l'agriculture nécessite une surveillance constante du champ afin de réagir au moment opportun. De ce fait, il est impossible de généraliser les problèmes et de suggérer une solution commune. Par conséquence, la collecte des données en temps réel peut nous aider à avoir des détails complexes sur chaque problème et par la suite de proposer la meilleure solution pour ce cas particulier.

L'application de l'intelligence dans le domaine agricole a commencé dès les années 1983 [17]. Afin de résoudre les problèmes rencontrés dans l'agriculture, plusieurs techniques ont été proposées (Figure 1) comme dans la gestion des cultures, la lutte contre les parasites, la gestion des sols, le contrôle de l'irrigation, etc. Ces techniques sont une combinaison entre l'application des algorithmes d'apprentissages pour l'extraction des informations nécessaires d'après les données collectées et le contrôle des équipements électroniques qui agissent sur le champ.

La croissance de la demande en produits alimentaires exerce une pression énorme sur les producteurs agricoles, particulièrement en raison de l'augmentation de la population, de la réduction de la superficie des terres arables et de la pénurie de certaines ressources. Pour répondre à cette demande, les agriculteurs ont souvent recours à l'utilisation aléatoire d'engrais et d'autres produits chimiques pour augmenter la production agricole. Cependant, cette surutilisation, qui ne respecte pas forcément les principes de durabilité écologique, a des effets négatifs sur la santé des sols et, par conséquent, sur la santé des plantes [16].

La nature dynamique de l'agriculture nécessite une surveillance constante des champs afin de réagir au moment opportun. Il est donc impossible de généraliser les problèmes et de proposer une solution unique. La collecte de données en temps réel

peut fournir des détails complexes sur chaque problème, permettant ainsi de proposer les meilleures solutions pour chaque cas particulier.

L'application de l'intelligence dans le domaine agricole a commencé dès les années 1983[17]. Pour résoudre les problèmes rencontrés en agriculture, plusieurs techniques ont été proposées, notamment dans la gestion des cultures, la lutte contre les parasites, la gestion des sols et le contrôle de l'irrigation (Figure 1.1). Ces techniques combinent l'application d'algorithmes d'apprentissage automatique pour extraire les informations nécessaires à partir des données collectées et le contrôle des équipements électroniques qui agissent sur le champ.

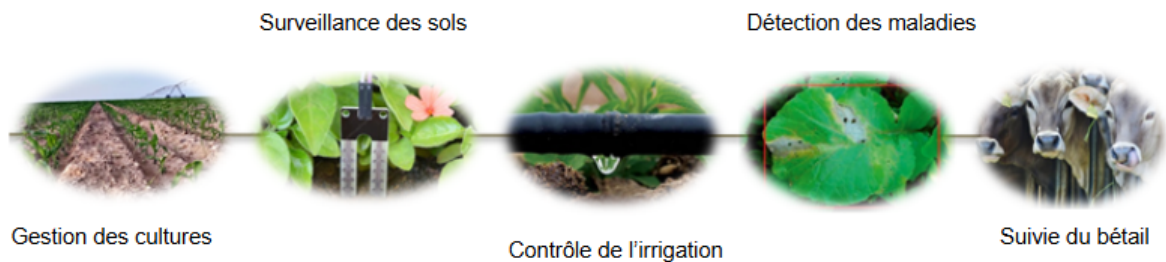


FIGURE 1.1 – Techniques de la PA.

1.2.1.1 Gestion des cultures

Avant d'élaborer des plans de gestion efficaces des pratiques agronomiques sur les cultures, il faut bien étudier des indicateurs de leurs rendements comme l'indice de surface foliaire (Leaf Area Index, LAI) (Figure 1.2) et la biomasse [18]. Ces derniers sont essentiels pour le suivi de la santé, la qualité et le développement des cultures et peuvent être exploités pour avoir des prévisions sur leur croissance.

Les anciennes méthodes physiques et optiques d'estimation du LAI comme par exemple les méthodes de terrain destructives sont gourmandes en temps et exigent beaucoup de main-d'œuvre en offrant peu de détails sur la variabilité spatiale de la croissance et de la biomasse des cultures [19]. En revanche les méthodes de télédétection comme par exemple, les images hyper-spectrales [20], peuvent contribuer dans l'obtention des informations pertinentes sur les caractéristiques spécifiques au site (comme, les sols, la topographie), la gestion (comme, l'eau, les nutriments et

autres intrants) et divers facteurs de stress biotiques et abiotiques (comme, les maladies, les mauvaises herbes, le stress hydrique et nutritif). De telles informations sur les conditions des cultures et les pratiques de travail du sol peuvent aider à développer des plans de gestion spécifiques au site, y compris l'application variable d'eau, de nutriments et de pesticides pour augmenter l'efficacité de la production et de la gestion. En règle générale, ces études utilisent un ensemble de données de référence (par exemple, le LAI mesuré et les indices de végétation correspondants) pour développer une approche basée sur la régression ou l'apprentissage automatique pour estimer le LAI et/ou la biomasse pour un champ cible [21].



FIGURE 1.2 – Appareil de calcul de LAI [22].

1.2.1.2 Contrôle de l'irrigation

L'eau est ce qui maintient le monde en vie, même si elle semble être la ressource la plus abondante sur terre, mais en réalité, les statistiques de l'organisation des nations unies pour l'alimentation et l'agriculture (FAO) [23] montrent que l'eau douce ne représente que 2,5 % de toute l'eau terrestre dans le monde. Les 97,5 % restants ne sont pas utilisables par l'homme car ils se trouvent sous forme d'eau de mer ou d'eau saumâtre, de neige ou de glace. Cependant, l'extension du processus d'irrigation dans l'agriculture est l'un des principaux moteurs de l'intérêt mondial croissant pour l'eau [24], car elle consomme 70 % des prélèvements d'eau douce dans le monde par rapport

à d'autres secteurs tels que le domestique et l'industrie [25]. Par conséquent, pour résoudre ce problème, il était nécessaire d'intégrer l'intelligence dans l'irrigation, car le processus d'irrigation automatisé peut assurer les besoins en eau des cultures, mais les technologies nouvellement adoptées, en plus d'arroser les cultures, prennent en considération plusieurs contraintes telles que comme la variabilité des facteurs temps, lieu, et type de culture. Par exemple, la tomate nécessite des tuyaux d'irrigation (Figure 1.3) enterrés [26] tandis que la betterave à sucre [27] nécessite des rampes d'irrigation localisées. Une autre contrainte est la quantité d'eau nécessaire qui change d'une plante à l'autre. D'après certains travaux, l'utilisation de robots de petites tailles aiderait à réduire la fréquence de labourage nécessaire à la surface agricole. il est aussi possible pour les robots de percer et placer les graines ou les semis sans avoir à labourer le sol, ou bien optimiser le processus du labourage en faisant des micro-labourages autour de l'endroit du semis, qui a le plus d'influence sur la croissance de la plante.



FIGURE 1.3 – Contrôle de l'irrigation par tuyaux [28].

Savoir où, quand et combien irriguer permet de minimiser les pertes de rendement dues au stress hydrique et de préserver l'eau en retour. L'irrigation de Précision (Precise Irrigation, PI) est un axe très important dans la PA. Contrairement aux technologies d'irrigation en boucle ouverte (sans rétroaction), qui fonctionnent selon un programme et des minuteries déjà préprogrammés [29]. PI (irrigation de surface ou enterrée) surveille les conditions météorologiques, les conditions du sol, l'évaporation

et l'exploitation de l'eau des plantes [30] pour ajuster automatiquement le programme d'arrosage en fonction des conditions réelles des terres agricoles. De plus, cela améliore considérablement l'efficacité de l'utilisation de l'eau.

Cependant, les besoins en eau peuvent être déterminés par différents facteurs tels que la disponibilité de l'eau dans le sol, la quantité de pluie, etc., dont il existe plusieurs approches pour les quantifier telles que l'humidité du sol [31] ou l'évapotranspiration [32]. Ainsi, les systèmes d'irrigation autonomes utilisent des algorithmes d'apprentissage automatique pour traiter les différentes données quantifiées par des techniques de surveillance en temps réel, pour extraire des informations et par la suite prendre des décisions de contrôle de l'irrigation ou pour faire des prévisions sur les besoins en eau dans le futur. Différentes techniques de surveillance en temps réel peuvent être appliquées dans la PI afin de détecter des informations utiles pour l'irrigation, telles que le calcul de l'évapotranspiration par télédétection [33, 34], la PI en utilisant des robots [35] et un système d'irrigation intelligent basé sur l'internet des objets (IoT), en utilisant des WSNs [5].

1.2.1.3 Surveillance du sol

La PA vise à optimiser la production de rendement en surveillant l'opération d'irrigation, l'application d'engrais, de pesticides et la croissance des cultures à l'aide des mesures directes ou indirectes de plusieurs variables du sol qui donnent des informations immédiates sur les besoins des cultures et de leurs milieux de croissance [36] comme, la température sol air, l'humidité (Figure 1.4), le carbone organique du sol, etc. Ces facteurs varient dans l'espace et dans le temps (pendant et entre les saisons) en raison des différences de sols, de gestion, de topographie, de conditions météorologiques et d'hydrologie [37].



FIGURE 1.4 – Système de surveillance de l’humidité du sol [38].

A l’époque, afin de consulter manuellement ces variables du sol, plusieurs voyages doivent être effectués, comme par exemple, il est difficile de suivre manuellement la température et les niveaux d’humidité du sol car la mesure manuelle des points de données clés sur les cultures est souvent difficile, prend du temps et plus susceptibles d’être inexactes.

De ce fait, pour faire face à ce problème, nous pouvons intégrer des nouvelles technologies comme les capteurs à distance montés sur un tracteur (par exemple, Green Seeker, Yara N-sensor, et Crop Circle). Cette technique peut offrir des informations en temps réel sur l’état des éléments nutritifs des plantes à des taux d’engrais variables dans l’espace. Ces capteurs qui sont généralement plantés sur la rampe de pulvérisation, collectent des données utiles pour le calcul de plusieurs indices de végétation (comme, Normalized Difference Vegetation Index, NDVI et Soil Adjusted Vegetation Index, SAVI), qui de leur part sont en corrélation avec les caractéristiques des plantes comme la teneur en chlorophylle, l’activité photosynthétique, et la productivité des plantes [39].

De plus, d’autres méthodes ont prouvé leur utilité dans la surveillance des sols comme les images capturées par la technologie des drones [40]. Ces images peuvent déterminer l’adéquation de plusieurs indices de végétation et la hauteur des cultures en précisant les taux d’application d’engrais en cours de saison pour les cultures. Un autre type d’image qui est les images satellitaires multi-temporelles (par exemple, les images prises de RapidEye [41]), peut développer des cartes de la matière organique du sol à l’échelle du terrain pour aider à la gestion de précision spécifique au site.

Cependant, les WSNs peuvent de leur part collecter des données en temps réel sur les caractéristiques des sols, des précipitations, de la température et d’autres mesures

au fil du temps pour suivre les tendances, prévoir les besoins d'irrigation, le taux exact des fertilisants et communiquer par la suite ces informations aux stations de base pour le traitement des données et la prise de décision en utilisant la technologie IoT, ce qui permet aux agriculteurs d'avoir les meilleurs plans de gestion des sols dans les meilleurs délais [42].

1.2.1.4 Détection des maladies

Dans la nature, les cultures sont souvent menacées par des maladies qui peuvent affecter négativement la production agricole et les profits agricoles. Donc toutes les détections précoces de ces maladies liées aux plantes et leur étendue spatiale peuvent sûrement aider à stopper sa propagation et réduire les pertes de production.

Parmi les anciennes méthodes de détection des maladies on trouve le repérage sur terrain [43], cette technique est lente car elle ne peut pas détecter la maladie dans ses premières phases, et nécessite beaucoup de main d'œuvre ce qui peut entraîner beaucoup d'erreur dans la détection [44]. Par contre, les technologies de la PA ont été très efficaces dans la lutte contre les maladies des plantes. Ces techniques comme la télédétection (par RVB, imagerie multi-spectrale, hyper-spectrale, thermique (Figure 1.5) et de fluorescence) permettent de diagnostiquer les maladies par leurs premiers symptômes. Le principe de fonctionnement de ces techniques est bien d'élaborer la corrélation entre la maladie des rayures foliaires de la vigne et le NDVI généré à partir des images [45].

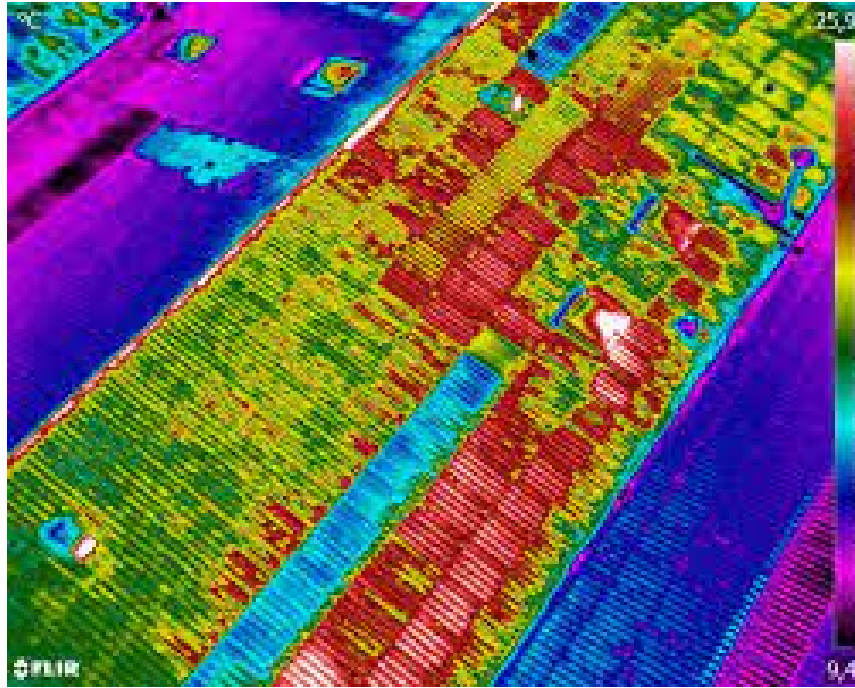


FIGURE 1.5 – Détection des maladies en utilisant des images thermique [46].

Une autre technique qui se présente comme très efficace dans la détection des nuisibles est bien les WSNs qui intègrent des capteurs acoustiques, ces derniers peuvent détecter la présence des nuisibles en utilisant leur niveau sonore produit, en le comparant avec un seuil défini. Cette technique informe l'agriculteur sur la zone précise où se produit l'infestation. En conséquence, l'agriculteur ne doit pas consulter et ajouter des produits chimiques dans toutes les parties de la parcelle mais seulement les lieux infectés [47].

1.2.1.5 Suivre du bétail

Afin de garantir une production suffisante de nourriture et d'animaux pour la population croissante, les plans de gestion de la production devraient être élargis pour inclure également une bonne maîtrise de l'élevage du bétail, c'est ce qu'on appelle Élevage de Précision (Precision Livestock Farming, PLF). Cette réglementation repose sur la surveillance automatique des animaux individuels, son application peut être bénéfique pour le suivi de la croissance des animaux, améliorer la production laitière, prévenir les maladies en poursuivant le comportement des animaux et permettre également de surveiller leur environnement physique [48].

La PLF utilise différentes méthodes de détection comme des caméras à technologie de capteur [49], des microphones en les combinant avec des moyens de communication comme des outils de communication sans fil et des connexions internet et avec l'utilisation du cloud pour le stockage [50]. Ces méthodes permettent aux agriculteurs de prendre des décisions en temps réel. Par exemple, dans le cas de la surveillance du comportement de pâturage parmi les variables détectées et qui ont un impact sur la prise de décision, on trouve le suivi du comportement des animaux comme leurs vitesses de déplacement, les modes de pâturage, les chemins et les zones privilégiées [51].

D'une autre part, dans l'industrie de l'élevage (Figure 1.6), la santé animale est très importante car elle a un impact direct sur la bonne croissance des animaux. La maladie des animaux peut altérer la production, le bien-être des animaux par la douleur et l'inconfort et même causer leur mortalité [52]. De plus, ces pathologies peuvent même atteindre les êtres humains dans les zones où les animaux sont proches d'eux. De ce fait, la prédiction des problèmes de santé dans leurs premières phases est une des solutions proposées par la PLF [53].



FIGURE 1.6 – Surveillance en élevage [54].

Nous prenons comme exemple la maladie de la mammite chez les vaches, le traitement des données collectées par des capteurs d'air en utilisant des algorithmes d'apprentissage précis permet de prévenir les agriculteurs de cette maladie. Dans l'industrie des volailles, les capteurs d'air assurent le suivi d'une manière constante la concentration de composés organiques volatils dans l'air causée par l'augmentation du nombre d'oiseaux infectés, ce qui permet de prédire l'apparition de la coccidiose [55]. De même, l'analyse des images captées permet aussi de créer un modèle pour détecter des boîtes basées sur la locomotion animale [56].

1.2.2 Avantages de la PA

La population croissante qui est estimée à être plus de 10 milliards vers les années 2050, nécessite de nouveaux plans et réglementations pour assurer la production alimentaire en garantissant des produits agricoles de qualité et un stock sain de bétail tout en préservant le maximum des ressources naturelles [57].

Pour répondre à ce problème, les chercheurs ont proposé d'intégrer la technologie dans le domaine agricole après avoir bénéficié de ses avantages dans d'autres domaines comme la médecine, la sécurité, etc. La chose qui a vu le jour dans les années 1980 en faisant naître la PA [58]. Cette approche a permis de faire une révolution dans le domaine agricole. Premièrement, la PA permet d'avoir un gain économique. Deuxièmement, la variété intra-parcelle est la variété des besoins des cultures dans une même parcelle nécessite des moyens de surveillance en temps réel des cultures [59]. A l'époque, l'agriculteur ne faisait pas attention à ce sujet, Il procédait à l'irrigation et à l'ajout de fertilisants de manière uniforme sur toutes les parcelles, ce qui a eu un impact négatif sur la production et le rendement des terrains. Par contre, avec l'intégration des technologies comme les WSNs et les techniques de télédétection, l'agriculteur a plus de détails sur les besoins exigés des cultures tout en ayant des informations sur la localisation des zones d'intérêt en utilisant des systèmes GNSS. De leur part, les robots agricoles ont présenté des avantages majeurs dans la collecte automatique de cultures, la plante des graines, la distribution de l'eau et même dans la distribution d'engrais [60] naturel aux plantes. De plus, ces mêmes machines peuvent être les meilleurs supports pour des capteurs de télédétection qui permettent d'avoir

les caractéristiques du sol planté. Cependant, ces techniques peuvent même prévenir les agriculteurs dans le cas de la présence de maladies et des nuisibles dans les zones d'intérêts la chose qui permet aux concernés de suivre le cycle de développement des cultures depuis la phase de pré-plantation des grains jusqu' à la collecte des cultures.

En outre, en ce qui concerne le suivi de l'élevage, l'application du PLF dans l'élevage du bétail a permis de protéger les êtres humains et les animaux des dangers des maladies infectieuses, comme dans le cas de la maladie de mammite chez les vaches [55]. Ces techniques permettent le suivi automatique des animaux d'une façon individuelle, ce qui permet par la suite d'assurer une bonne croissance des animaux, des produits laitiers de qualité, et la détection de maladies. Ces dernières permettent également d'étudier le comportement animal et d'agir par la suite sur l'endroit où ils vont grandir.

1.2.3 Outils et technologies pour la PA

Après la révolution technologique que subit le monde ces dernières années, on peut constater un intérêt croissant pour l'application des différents outils technologiques dans l'agriculture comme dans plusieurs domaines de la vie. Dans cette section, nous allons présenter les différentes techniques qui ont été appliqué dans la PA et ont prouvé un avantage considérable par rapport aux méthodes classiques de l'agriculture. Nous avons réparti ces techniques en deux groupes (Figure 1.7), le premier est pour les outils d'intervention sur le champ comme les robots agricoles et les machines de la distribution différenciée des engrais et des produits phytosanitaires, et le deuxième est pour les outils de collecte des données comme la télédétection par les image satellitaires ou par les images des drones, et les WSNs.

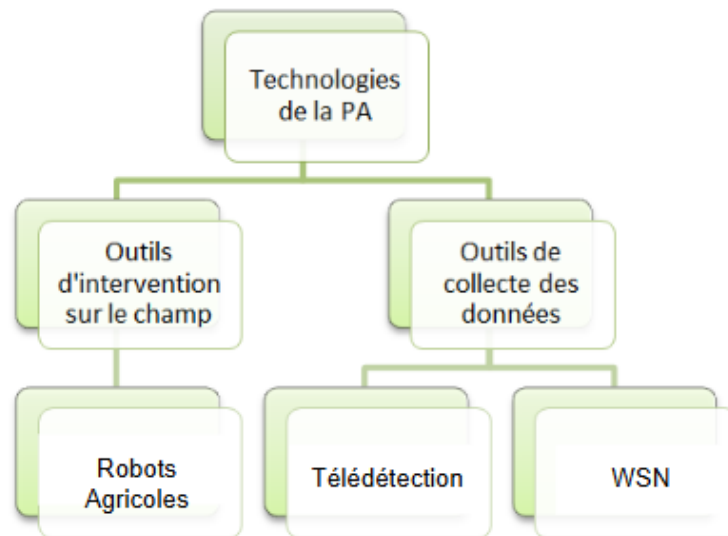


FIGURE 1.7 – Répartition des technologies utilisées dans la PA.

1.2.3.1 Outils d'intervention dans les champs (les robots agricoles)

Les processus agricoles sont définis comme des chaînes de tâches systématiques et répétitives qui dépendent du temps, avec quelques processus qui diffèrent selon le type de culture (culture permanente ou culture arable). Par conséquent, afin de remplacer le travail humain pour réaliser ces tâches plusieurs technologies peuvent faire partie comme les robots agricoles. On peut les définir comme des machines robotiques mobiles qui sont utilisées pour réaliser des tâches précises dans les processus agricoles. Ces machines peuvent performer pour plusieurs types d'activités, comme la production animale, forestière, et dans la production agricole des cultures qui va de la sélection végétale à l'établissement, la culture et la récolte des cultures [61].

Parmi les applications des robots agricoles, on trouve la cartographie des semences, où les robots agricoles ont montré un avantage majeur, car en utilisant un module GPS, les robots peuvent passivement enregistrer la position géo-spatiale de chaque graine au fur et à mesure qu'elle s'enfonce dans le sol (Figure 1.8). Son principe de fonctionnement général repose sur la coupure du faisceau infrarouge du capteur infrarouge lorsque la graine tombe. A ce moment-là, le robot déclenche un enregistreur de données qui enregistre la position et l'orientation du semoir [62].



FIGURE 1.8 – Plantation des grains avec un robot agricole [63].

Un autre type d'application des robots agricoles est bien le contrôle du désherbage, cette étape est connue comme l'une des activités les plus répétitives, fastidieuses et chronophages du processus de production agricole. De ce fait, les robots agricoles peuvent intervenir pour résoudre ce problème par deux méthodes, le désherbage mécanique [64] et le désherbage chimique [65]. Cependant, la mécanique procède à la destruction des plantes adventices, principalement en les arrachant, en les brûlant ou en les coupant. Tandis que le désherbage chimique procède à la pulvérisation d'herbicides qui ont un effet toxique sur les plantes adventices.

De plus, les robots peuvent être une bonne solution pour la détection des maladies des cultures, en intégrant un système basé sur la vision comme des caméras couleur car ce problème est principalement une tâche visuelle. Cette approche a permis de détecter l'oïdium et le virus de la flétrissure tachetée de la tomate avec une précision de 80 % et de 61 % pour ces deux maladies respectivement [66, 67]. De plus, ces machines intelligentes ont été utilisées dans d'autres types d'application comme la pulvérisation, le repérage des cultures, la récolte, etc. Dans le tableau 1.1, nous allons présenter quelques exemples des travaux de la littérature qui ont bénéficié des compétences des

robots agricoles dans l'agriculture. Ce tableau définit les travaux, la culture plantée, le type d'application et la machine utilisée.

1.2.3.2 Outils de collecte des données

La PA repose sur la collecte de données détaillées pour optimiser la gestion des ressources et améliorer les rendements. Voici quelques outils couramment utilisés pour la collecte de données dans ce domaine.

i. Télédétection

Sans avoir besoin d'être en contact direct avec les plantes, la télédétection permet d'obtenir des informations sur leurs états à l'aide des ondes électromagnétiques capter par un capteur passif (dans le cas général) ou bien par des images satellitaires (Figure 1.9) ou des images captées par les drones. Sachant que la réflectance des plantes change selon différents facteurs, la télédétection peut être utilisée dans la prévision du rendement pour la récolte, dans l'évaluation des besoins en eau pour l'irrigation et en nutrition des plantes pour la fertilisation et pour la détection des dommages affectant la culture [79]. Les systèmes de télédétection appliqués dans la PA sont généralement classés selon le type des capteurs utilisés et la plateforme des capteurs. Les types sont généralement des capteurs passifs ou bien actifs, la différence entre eux et bien dans la manière de détecter l'information. Pour les capteurs actifs, ils émettent leurs propres radiations et détectent la réflexion de ceux-ci par l'objet d'intérêt. Par contre, les capteurs passifs, captent, convertissent, détectent les radiations incidentes, émises ou réfléchies par l'objet [80]. De plus, concernant les plateformes utilisées pour la télédétection on peut commencer par la télédétection en utilisant les satellites, cette plateforme offre une grande couverture spatiale ce qui permet l'extraction de données (sous formes des images multi spectrale) sur une longue période de temps avec des informations consistantes et comparables. Malheureusement, cette technique peut causer des problèmes de résolutions des images, du temps de revisite qui peut affecter négativement le processus d'abreuvement et la distribution des nutriments, d'autant plus que ces processus nécessitent une intervention rapide. De ce fait, une autre technique de

TABLE 1.1 – Exemples des travaux de la littérature qui ont bénéficié des compétences des robots agricoles dans l’agriculture.

Travaux	Culture plantée	Outils de perception	Type d’application
[68]	Blé	Capteur de force, capteur de déplacement, capteur d’angle	Cartographie d’ensemencement
[69]	Riz	Boussole, roue codeuse	Cartographie d’ensemencement
[64]	Maïs	Caméras, capteurs de distance optiques et acoustiques	Control de désherbage (chimique)
[70]	Betterave à sucre	Camera en couleur	Control de désherbage (mécanique)
[71]	Olivier	Deux caméras DSLR (une en mode BNDVI), une caméra multi spectrale, un système hyper spectral dans le domaine visible et NIR, une caméra thermique, lidar, un capteur IMU	Détection des maladies
[72]	Riz, tomate	Caméra RVB	Détection des maladies
[73]	Raisins	Caméra RVB et IR, détecteur de garde forestier laser, IMU, capteur de pression, etc.	Repérage des cultures
[74]	Sorgho	Caméra stéréo, caméra RVB avec lentilles fish eye, pénétromètre	Repérage des cultures
[75]	Concombre	Capteurs de choc, infrarouge Capteurs, capteurs à induction	Pulvérisation
[76]	Cultures maraîchères	Caméra hyperspectrale, vision stéréo, infrarouge thermique Caméra, caméra couleur monoculaire	Pulvérisation
[77]	Luzerne, soudan	Caméra couleur, gyroscope	Récolte
[78]	Pommier	Caméra couleur, caméra tridimensionnelle basée sur le temps de vol	Récolte

télédétection qui est l'utilisation des avions sans pilote, et plus récemment les drones (Unmanned Aerial Vehicle, UAV) [81] à qui, on a recours plus souvent durant ces dernières années. Ceci est grâce à leur coût moins élevé ainsi qu'aux avancées technologiques liées aux drones qui sont utilisés avec des caméras qui captent des images dans le spectre visible, proche infrarouge, infrarouge thermique, auquel on fait référence comme les Systèmes Aériens sans Pilote (Unmanned Aerial System, UAS) [82].

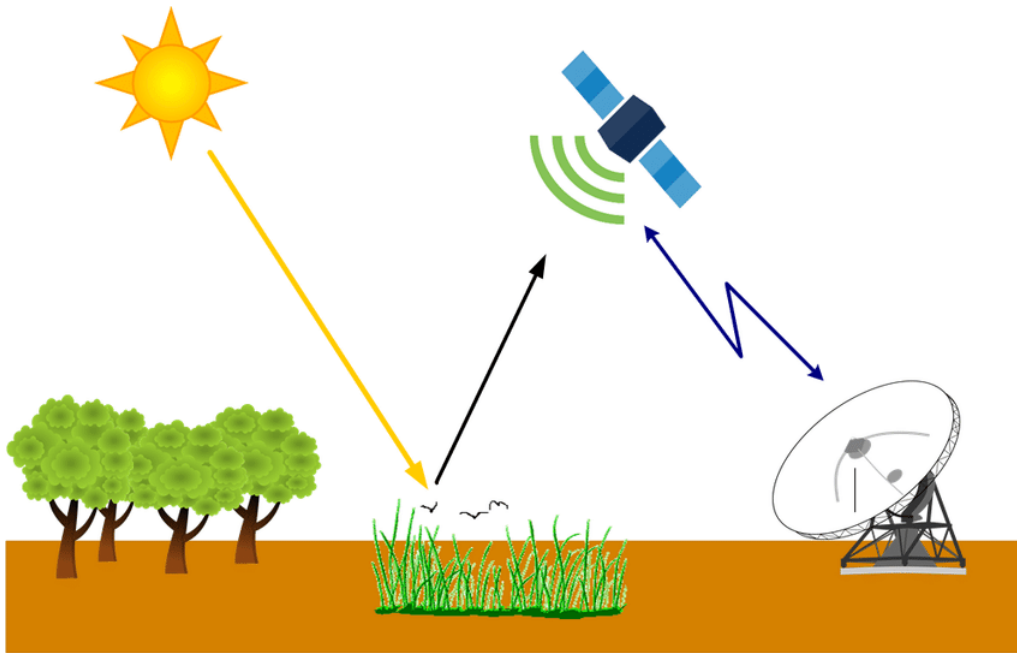


FIGURE 1.9 – Exemple d'un système de télédétection satellitaire [83].

L'utilisation des UAS permet d'obtenir des images à haute résolution, permettant un grand nombre d'applications liées à la PA. Leurs désavantages restent le temps de vol court et la capacité limitée à supporter du poids. On a aussi la détection à distance au sol qui se fait à l'aide de capteurs à main ou bien des capteurs embarqués sur des tracteurs agricoles qui sont utiles lorsqu'il s'agit d'une petite surface d'intérêt [84]. Pour la détection d'un stress biotique et d'un stress abiotique ou pour une calibration d'autres mesures basées sur d'autres plateformes.

La nécessité de télédétection des sols et des terres est reconnue depuis longtemps (depuis les années 1958 [85]) afin d'assurer une gestion durable des ressources naturelles à l'échelle locale, régionale et nationale [86]. Si on connaît les vrais

besoins de la terre sur l'échelle physiques, biologiques et chimiques, on pourra réaliser de bonnes stratégies pour la programmation de l'irrigation, le drainage, la distribution des nutriments et les autres besoins des cultures. Cette technique permet d'avoir des informations utiles pour surveiller la santé du sol et de la végétation, ainsi que les paramètres hydrologiques et climatiques, qui sont importants pour la PA (par exemple, le carbone organique du sol, l'humidité du sol, le NDVI, le LAI, les eaux souterraines et les précipitations).

Parmi les applications de la télédétection dans la PA, on trouve le contrôle de l'irrigation en prenant en compte la variabilité au sein du champ, cette technique permet de discerner cette variabilité ce qui permet par la suite d'appliquer une irrigation à débit variable selon les besoins en eau avec des systèmes d'irrigation couramment utilisés tels qu'un pivot central. En utilisant les images collectées par les plateformes de télédétection plusieurs fois au cours d'une saison de croissance, on pourra déterminer divers indicateurs de la demande en eau de cultures telles que l'évapotranspiration (Évapotranspiration, ET) [39], l'humidité du sol et le stress hydrique des cultures. Par conséquent, ces indicateurs aident à estimer les besoins en eau des cultures. Par exemple, depuis les images optiques, on peut développer le NDVI et le SAVI, qui peuvent être utilisés pour diagnostiquer le stress hydrique et les conditions d'humidité du sol pour de nombreuses cultures.

Concernant la distribution des nutriments, les indices calculés par les données de télédétection comme le (par exemple, NDVI et SAVI) permettent de définir plusieurs paramètres des plantes comme, la teneur en chlorophylle, l'activité photosynthétique et la productivité des plantes. Par conséquent, la cartographie de ces indices peut ainsi aider à comprendre la variabilité spatiale de l'état des éléments nutritifs des cultures, ce qui est important pour la PA [87].

D'une autre part, la télédétection a pu s'avérer sa bonne utilisation dans la détection des maladies des plantes. En utilisant les images, plusieurs techniques comme l'imagerie RVB, multi spectrale, hyper spectrale, thermique et de fluorescence ont présenté leurs avantages dans la détection des maladies dans leurs premières étapes [88]. De même pour la surveillance des cultures, la télédétection

peut estimer le LAI et la biomasse pour une variété de cultures, ces deux indices sont des indicateurs essentiels de la santé et du développement des cultures [89]. Le tableau 1.2 suivant représente les outils de perceptions utilisées pour la télédétection selon le type d'application dans la PA.

TABLE 1.2 – Exemples des travaux de la littérature qui ont bénéficié des compétences de la technologie de la télédétection dans l’agriculture.

Travaux	Culture plantée	Outils de perception	Type d’applica- tion
[90]	Blé	Satellite Landsat 1	Gestion des cultures
[91]	Riz	Satellite Landsat 8	Gestion des cultures
[92]	Maïs	Satellite Spot 2	Water manage- ment
[93]	Betterave sucre	à Satellite IRS 1A	Control de l’irri- gation et la ges- tion des nutri- ments
[94]	Olivier	Satellite Lidar	Topographie et la gestion des nutriments
[95]	Riz, tomate	Satellite IKONOS	L’estimation de la valeur ET
[96]	Raisins	Satellite ECOSTRESS- phytir	L’estimation de la valeur ET
[97]	Sorgho	Satellite quickbird	Détection des maladies
[98]	Concombre	Satellites SPOT-6 et SPOT- 7	Détection des maladies
[89]	Cultures maraîchères	Satellite Radarsat-2	LAI et Biomasse

ii. Réseaux de capteurs sans fils (WSN)

La technologie WSN utilise un groupe de capteurs sans fil pour collecter, surveiller et analyser les données détectées [99], de sorte que dans le domaine de

l'agriculture, un WSN peut optimiser la qualité des cultures et préserver les ressources naturelles grâce à l'utilisation des capteurs déployés qui surveillent la valeur vitale des paramètres physiques (tels que l'humidité, la température et le niveau de PH, etc.) pour définir les besoins exacts des plantes. Par ailleurs, il existe plusieurs types de WSN en agriculture, qui sont classés selon leur mode de déploiement en deux classes, la première classe est les Réseau de Capteurs Sans Fil Terrestre (Terrestrial Wireless Sensor Networks, TWSN) où les capteurs sont déployés au-dessus du sol (Figure 1.10) et la seconde est les Réseau de Capteurs Sans Fil Souterrain (Underground Wireless Sensor Network, UWSN) où des capteurs sont déployés à l'intérieur du sol.

Le premier mode présente des fonctionnalités plus puissantes par rapport au second telles que le faible coût, la faible consommation d'énergie et une portée de communication élevée pouvant aller jusqu'à 100 mètres [100]. Cependant, le choix entre les deux modes est influencé par le type d'application des WSNs en agriculture.



FIGURE 1.10 – Exemple d'un TWSN appliqué à la PA [101].

Chaque WSN est composé d'un ensemble de nœuds de capteurs et de stations de base qui travaillent ensemble pour détecter, transmettre, traiter et interpréter les informations. De plus, chaque nœud capteur est composé d'une unité de traitement, une unité de stockage, une source d'alimentation, un capteur et un module de communication. Les composants des nœuds capteurs sont définis en fonction de leur domaine d'application. Par exemple, la technologie de communication la plus adoptée en agriculture est la technologie Zigbee grâce à ses propriétés

telles que la portée de communication qui varie entre 10 et 100 m, et sa faible consommation d'énergie [102, 103]. Une fois déployés, les nœuds coopèrent entre eux d'une manière autonome afin de collecter et de transmettre des données vers une station de base dans le but de surveiller et/ou de contrôler un phénomène donné. Dans [104], l'utilisation de l'agriculture intelligente a conduit à une augmentation du rendement de 1,75 %, une baisse de 17 à 32 dollars des coûts énergétiques par hectare et la préservation de 8 % de l'eau utilisée.

La technologie WSN a montré de très bonnes performances en agriculture avec plusieurs types d'utilisation comme le suivi des animaux [105], la surveillance des sols pour la gestion précise des ressources [106], la surveillance des cultures pour la prédiction du moment adéquat de la récolte [79], l'utilisation des engrais et le contrôle de l'irrigation [107].

Les applications à base des WSNs dédiées à la PA peuvent assurer plusieurs tâches à la fois en ajoutant d'autres nœuds à l'architecture de base. Par exemple : les WSNs utilisés pour le contrôle des conditions d'environnement de la culture à base des informations tirées du sol, peuvent être aussi utilisées pour la prédiction des états et de la qualité des cultures produits. Mais dans ce cas-là de nouveaux problèmes vont être présents comme : la détermination optimale de la stratégie de déploiement, les protocoles de routage, les limites des intervalles de mesure, et l'accès au support économe en énergie.

Les WSNs ont récemment évolué vers IoT friendly-WSN, en adoptant des standards plus génériques en termes de communication, permettant un accès à distance à internet et en mettant en œuvre des algorithmes intelligents de méta traitement des données visant à améliorer la surveillance et/ou le contrôle des parcelles agricoles [108]. Comme dans le cas des serres, ou dans de nombreux cas, cela nécessite une plus grande précision en termes de surveillance et de contrôle car il a été démontré que les serres présentent une variabilité climatique importante, qui affecte la productivité des plantes, voire leur nuit [109]. De plus, un environnement optimal qui absorbe les conditions climatiques extrêmes qui ont un effet négatif sur la productivité des animaux est une question sérieuse. De ce fait, plusieurs applications WSNs ont été réalisées dans les domaines de l'élevage, car cette technologie assure non seulement le contrôle et le suivi des

conditions climatiques, mais elle permet aussi d'améliorer les pratiques d'alimentation, l'instrumentation et l'analyse du comportement animal [110]. Le tableau 1.3 présente des exemples des travaux de la littérature qui ont bénéficié des compétences de la technologie de WSN dans l'agriculture.

1.3 Architecture de base d'un WSN

Un WSN est composé d'un grand nombre de nœuds déployés dans une région d'intérêt, ajoutés à ceux-là, des nœuds puits ainsi qu'une ou plusieurs stations de base (voir figure 1.11) [123].

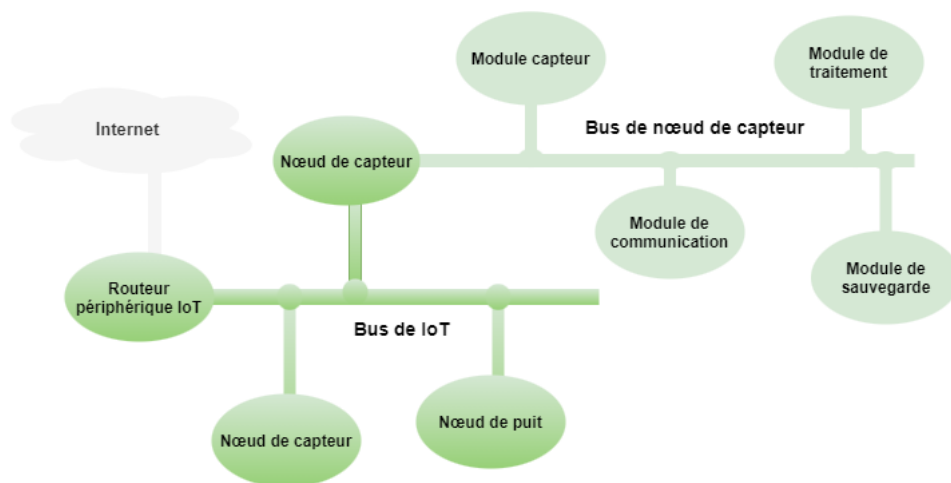


FIGURE 1.11 – Modèle architectural d'un WSN.

Station de base : c'est un point centralisé de contrôle dans le réseau. Elle est employée pour extraire des informations du réseau et envoyer des informations de contrôle aux différents nœuds du réseau.

Nœud capteur : les nœuds capteurs recueillent les informations liées à leur environnement de déploiement. Il existe deux types de nœud capteur : les capteurs fixes et les capteurs mobiles. Un nœud capteur comme le présente la Figure 1.12, est généralement composé des cinq modules suivants :

- Module d'acquisition des données ;
- Module de traitement ;
- Module de stockage ;
- Module de communication ; d'énergie.

TABLE 1.3 – Exemples des travaux de la littérature qui ont bénéficié des compétences de la technologie WSN dans la PA.

Travaux	Culture plantée	Outils de perception	Type d'application
[111]	Vergers d'orangers	Algorithme OASNDFA Nœuds de capteurs	Suivi des cultures
[112]	Sucre de canne	Capteur N pour mesurer les besoins en azote de la culture et récepteur GPS où les capteurs sont connectés	Suivi des cultures
[113]	Culture de pomme de terre	Nœuds de capteurs Capteur MTS400, cartes MDA300, protocole APTEEN	Suivi des cultures
[114]	Pomme de terre	Capteur d'humidité du sol Capteur de température GPS	contrôle de l'irrigation
[115]	Vignoble	Technologie de réseau de capteurs et d'actionnement sans fil (WSAN)	Contrôle de l'irrigation
[116]	Domaine agricole	Système de positionnement global Communication radio sans fil Bluetooth	Contrôle de l'irrigation
[117]	Domaine agricole	Actionneur de vanne Capteurs d'humidité du sol ZigBee	Commande de vanne d'irrigation
[118]	Vergers de pêchers	Capteur Arduino DS18B20 PHP Mysql Potentiel hydrique du sol	Surveillance et gestion de l'irrigation
[119]	Champ de blé	Capteurs de sol Capteurs colorimétriques Transistor à effet de champ	Mesurer et surveiller les attributs physiques et biochimiques du sol
[120]	Domaine agricole	Plateforme de capteurs mobile VERIS capteur vis-NIRS	Mesure le carbone organique présent dans le sol
[121]	Champ ouvert	Capteurs MEMS	Surveillance de la température du sol et de l'humidité
[122]	Coton	Thermomètre infrarouge Capteur CQIRT TN901	Mesurer les performances du terrain à l'aide des WSN

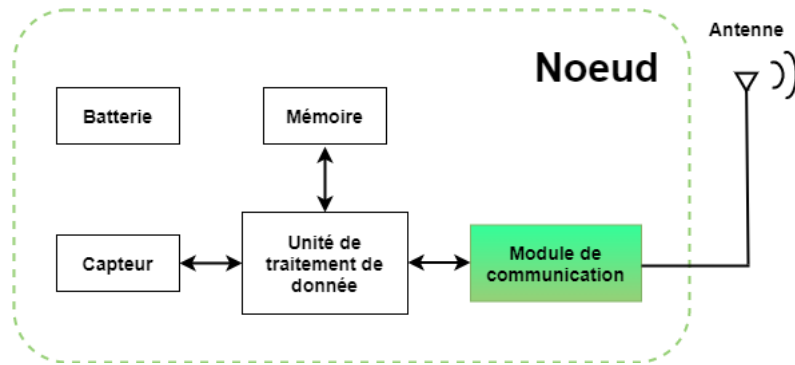


FIGURE 1.12 – Architecture de base d'un nœud capteur.

Nœud puits : aussi appelé Sink ou routeur, le Sink joue un rôle crucial en tant que point central de collecte et de traitement des données envoyées par les différents capteurs répartis dans le réseau[124]. Ce dispositif peut également fonctionner comme un nœud capteur mais il peut en plus recevoir, traiter et enregistrer des données provenant des nœuds capteurs. Il détient des capacités supérieures en termes de puissances de traitement, capacité de mémoire et autonomie d'énergie.

1.4 Éléments constructifs des WSNs

L'application des différentes technologies dans le domaine agricole a permis l'amélioration des méthodes traditionnelles de l'agriculture. De sa part, la technologie WSNs permet de faire la surveillance en temps réel des champs agricoles la chose qui permet par la suite d'agir et de prendre des décisions dans les meilleurs moments afin d'améliorer la qualité des produits. Donc, afin de mieux exploiter les bénéfices de cette technique, plusieurs paramètres doivent être bien choisis et sélectionnés pour la construction d'un WSN dédié à la PA. Dans les sous-sections suivantes nous allons présenter les différents modes et architectures de déploiements des WSN, les technologies de communications possibles, les topologies des WSNs et son alimentation.

1.4.1 Modes d'installation des WSN

Pour le domaine agricole, il existe deux modes d'installation des WSNs. Le TWSN et le UWSN.

1.4.1.1 Réseaux de capteurs sans fil terrestres (TWSN)

Dans ce mode d'installation des WSN, les nœuds sont placés au-dessus de la surface du sol. Ces différents nœuds sont interconnectés pour répondre aux exigences de l'application. Par exemple, dans le cas de smart irrigation, l'ensemble des nœuds capteurs placés sur le sol peut capter les valeurs exactes d'humidité et de température et d'envoyer ces informations à la station de base ou ces données seront traitées pour prendre des décisions de taux et de temps de l'irrigation en commandant à distance des pompes à eau [125].

1.4.1.2 Réseaux de capteurs sans fil souterrain (UWSN)

Le deuxième mode d'installation des WSNs est le réseau souterrain de capteurs sans fil UWSN. Dans ce mode, les nœuds sont plantés à l'intérieur du sol. De ce fait, les fréquences plus élevées subissent une atténuation sévère et les fréquences comparativement plus basses sont capables de pénétrer à travers le sol [126]. De plus, le rayon de communication devient limité et le réseau nécessite un nombre plus élevé de nœuds pour couvrir une vaste zone en raison d'une distance de communication comparativement plus courte. De plus, les nœuds sont connectés à un nœud passerelle qui se charge de transmettre les informations collectées à un puits de surface placé au-dessus du sol. Par la suite, les informations peuvent être transmises et utilisées comme dans le cas de TWSN. De plus, en raison de la distance de communication comparativement plus courte, un plus grand nombre de nœuds doivent être déployés pour être utilisés dans les WUSN. Le tableau 1.4 ci-dessous représente une comparaison entre les deux types d'installation selon plusieurs facteurs.

TABLE 1.4 – Comparaison entre les modes d'installation des WSN.

Caractéristiques	WSN Terrestre	WSN Souterrain
Installation	Placer sur la terre	Placer sous la terre
Profondeur	N'importe où sur la terre	la terre superficielle (0-30cm) et sous-sol ($> 30cm$)
Portée de communication	=100m	= 0.1-10m
Fréquence de communication	Haute (868/915 MHz, 2.4 GHz)	Faible (433 Mhz, 8-300 KHz)
Taille d'antenne	petite	grande
Consommation d'énergie	faible	haute
coût	faible	élevé

1.4.2 Architecture de déploiement des WSNs

Sur la base du mouvement des appareils et des nœuds en réseau, nous classons les architectures existantes dans les catégories suivantes :

1.4.2.1 Architecture stationnaire

Les nœuds doivent être déployés à une position fixe dans cette architecture et conserver la même pendant toute la durée de l'application. Car ils existent plusieurs applications où les nœuds doivent assurer des endroits typiques pour assurer le bon fonctionnement des applications comme dans le cas du système de gestion de l'irrigation, de la surveillance de la qualité des eaux souterraines et du contrôle de l'utilisation des engrais [127]. les WSNs stationnaires sont plus faciles à déployer et à entretenir. Les capteurs de nœuds peuvent être installés à un endroit optimisé et donc un plus grand nombre de nœuds peuvent être placés dans moins d'espace.

1.4.2.2 Architecture mobile

Dans l'architecture mobile, les différents composants du réseau changent leur position au fil du temps. Aussi appelée MANET (Mobile Ad hoc network), cette architecture peut être appliquée en utilisant des appareils mobiles qui s'auto-configurent, s'auto-réparent, s'auto-optimisent et s'auto-protègent à la fois. Selon le type d'application, plusieurs architectures peuvent être utilisées dans ce type, comme une architecture où les nœuds sont mobiles est la station de base reste fixe et une fois les nœuds sont près de la station où elle délivre les données collectées. Une autre architecture où la station de base change de position et collecte les données à partir de nœuds de capteurs stationnaires. On peut prendre comme exemple de cette architecture un réseau autonome de tracteurs et de téléphones portables transportant des agriculteurs au service d'opérations agricoles omniprésentes [128].

1.4.2.3 Architecture hybride

Les deux architectures précédentes peuvent être fusionnées dans une même application où on peut trouver des nœuds de capteurs qui sont stationnaires et des équipements agricoles qui changent de positions, des téléphones portables transportant des utilisateurs et des bovins en mouvement.

1.4.3 Technologies et normes de communication des nœuds dans les WSNs

Dans cette partie nous allons présenter les technologies de communication sans fil qui peuvent être utilisées et les WSNs pour des applications agricoles. De ce fait, nous nous intéressons à un présentateur de cinq technologies qui sont le Zigbee, Bluetooth, Wireless Fidelity (Wifi), General Packet Radio Service (GPRS)\3g \4g et le Worldwide Interoperability for Microwave Access (WiMAX).

1.4.3.1 Zigbee-IEEE 802.15.4

C'est la norme de réseau maillé sans fil la plus utilisée pour connecter les capteurs [129]. Dans le domaine agricole, la technologie ZigBee est préférée pour les applications

basées sur WSN car elle est fiable, économe en énergie et peu coûteuse. De plus, ZigBee peut communiquer des données à courte distance (10 à 20 m) sur des réseaux multi-niveaux, décentralisés, ad-hoc et maillés. Les appareils ZigBee compatibles ont un faible cycle de service, ce qui les rend idéaux pour les applications agricoles telles que la gestion de l'irrigation, le contrôle des pesticides et des engrais et la gestion de la qualité de l'eau, où des informations doivent être mises à jour régulièrement. Cependant, les applications ZigBee produisent de faibles débits de données de 20 à 40 kbps sur les fréquences 868/915 MHz et 2,4 GHz de la bande Industriel, Scientifique et Médical (Industrial, Scientific, and Medical, ISM). En règle générale, cette norme implique l'utilisation de matériel de faible qualité (comme un microprocesseur disposant de 50 à 60 Ko de mémoire) et inclut des méthodes de cryptage de sécurité.

1.4.3.2 Wifi

WiFi est une norme de réseau local sans fil (Wireless Local Area Network, WLAN) basée sur la famille de normes IEEE 802.11 (IEEE 802.11, 802.11a/b/g/n) pour l'échange d'informations ou la connexion sans fil à Internet. Il s'agit actuellement de la technologie sans fil la plus utilisée dans les appareils allant des téléphones intelligents et des tablettes aux ordinateurs de bureau et portables. La bande ISM 2,4 GHz offre un taux de transmission de données de 2 à 54 Mbps et une portée de communication acceptable de 20 m (intérieur) à 100 m (extérieur). Le WiFi élargit l'utilisation d'architectures hétérogènes dans les applications agricoles en connectant plusieurs types d'appareils sur un réseau ad hoc [130].

1.4.3.3 Bluetooth

Bluetooth, basé sur la norme IEEE 802.15.1, est une technologie sans fil à faible consommation et à faible coût qui permet une communication sur une courte portée (8 à 10 m) entre des appareils portables et des ordinateurs de bureau. La norme Bluetooth permet la communication en réseau personnel (Personal Area Network, PAN) sur la bande ISM 2,4 GHz. Les différentes versions du Bluetooth peuvent fournir un débit de données de 1 à 24 Mbps. Cette technologie est omniprésente, ce qui en fait une excellente option pour une variété d'applications agricoles. Bluetooth Low

Energy (BLE), qui a été introduit pour la première fois par Nokia en 2006 sous le nom de Wibree. Cependant, BLE et la principale norme Bluetooth version 4.0 ont fusionné en 2010. Pour réduire les interférences, le BLE utilise également la bande de fréquence ISM de 2,4 GHz avec sauts de fréquence adaptatifs. De plus, pour garantir la robustesse et l'authentification, BLE intègre sur tous les paquets une technique de cryptage CRC 24 bits et AES 128 bits. Les connexions un à un et un à plusieurs entre les appareils sont prises en charge par la topologie BLE [131].

1.4.3.4 GPRS \3g\4g

Le service de données par paquets pour les téléphones cellulaires GSM est appelé GPRS. Les systèmes 2G ont un débit de données de 50 à 100 kbps. Cependant, le débit et le délai en GPRS sont variables et dépendent du nombre d'autres utilisateurs utilisant la même ressource. Malgré le fait que le principal avantage du GPRS est de réduire la limite de portée des appareils sans fil. Si deux appareils se trouvent dans la même zone de service GSM, ils peuvent interagir. Cependant, il fonctionne mieux avec les applications de surveillance périodique qu'avec les applications de surveillance en temps réel. Le GPRS avancé, appelé Enhanced Data Rates for Global Evolution (EDGE), permet un débit de données accru dans les réseaux centraux GSM sans aucune modification matérielle ou logicielle. Les troisième et quatrième générations de technologies de communication mobile sont respectivement appelées 3G et 4G. En 3G et 4G, les taux de transfert de données correspondants sont de 200 kbps et 100 Mbps, respectivement [132].

1.4.3.5 WiMAX

WiMAX est l'acronyme de Worldwide Interoperability for Microwave Access, une norme de communication sans fil qui indique que la famille des normes IEEE 802.16 peut être utilisée ensemble. Sur les stations fixes, WiMAX peut atteindre un débit de données de 0,4 à 1 Gbit/s et une portée de transmission maximale de 50 km. Les débits de données du Mobile WiMAX varient entre 50 et 100 Mbps. De plus, par rapport à la 4G et à l'accès aux paquets à haut débit évolué (HSPA+), le WiMAX

consommerait moins d'énergie que le protocole Long-Term Evolution (LTE). Le WiMAX est la technologie la mieux adaptée aux applications agricoles impliquant la surveillance des actifs tels que la surveillance des systèmes agricoles, la surveillance des limites des zones de culture et le diagnostic en temps réel tels que le contrôle à distance des pompes à eau, des lumières et des portails, ainsi que le diagnostic à distance des systèmes agricoles. La prise en charge longue portée et les fonctionnalités de communication à haut débit le rendent la technologie la mieux adaptée [133]. Le tableau 1.5 suivant représente une comparaison entre les différentes technologies qui peuvent être utilisées par les WSNs pour une application agricole.

TABLE 1.5 – Technologies et normes de communication utilisées dans les WSNs pour une application agricole.

Paramètres	Zigbee	Wifi	Bluetooth	GPRS/3G/4G	WiMAX
Standard	IEEE 802.15.4	IEEE 802.11a,b,g,n	IEEE 802.15.1	-	IEEE 802.16a,e
Bande-de-fréquence	868/915 mhz, 2.4 ghz	2.4 ghz	2.4ghz	865 mhz, 2.4 ghz 2-66 ghz	
Débit-de-données	20-250 Kbps	2-54 Mbps	1-24 Mbps	50-100 Kbps/200 Kbps/0.1-1 Gbps	0.4-1 Gbps (station-naire), 50-100 Mbps (mobile)
Portée-de-transmission	10-100m	20-100 m	8-10 m	Zone de couverture-GSM-entière	<50 Km
Consommation-d'énergie	Faible	Haute	Moyenne	Moyenne	Moyenne
Coût	Faible	Elevé	Faible	Moyen	Elevé

1.4.4 Topologie des WSN

Un WSN peut être organisé selon différents types de topologie dont voici les principales.

1.4.4.1 Étoile (Star)

Le réseau est constitué d'un ensemble de nœuds directement reliés à un coordinateur central. Les nœuds ne sont pas autorisés à interagir entre eux et communiquent via le coordinateur central [134]. Cette topologie a pour avantages d'être facile à déployer, et de garantir des communications à faible latence entre les nœuds distants et la station de base [135]. L'inconvénient de cette topologie est que si le coordinateur central est défectueux, tout le réseau tombe en panne.

1.4.4.2 Arbre (Tree)

Le réseau contient un nœud racine, de nombreux routeurs et des périphériques d'extrémité. Tous les nœuds sont liés sous forme d'arborescence. Les nœuds d'extrémité sont directement liés au coordinateur et aux routeurs en tant que nœuds enfants. Un terminal ne peut interagir avec un autre terminal que via son nœud parent. L'inconvénient de la topologie arborescente est que si l'un des parents devient désactivé, les enfants de celui-ci ne peuvent pas interagir avec d'autres périphériques du réseau [134].

1.4.4.3 Maille (Mesh)

Une topologie maille est autogérée, c'est-à-dire que pendant la transmission, si l'un des chemins échoue, le nœud d'origine couvrira un chemin alternatif vers le nœud de destination. Tout appareil source peut interagir avec n'importe quel appareil de destination du réseau [134]. L'inconvénient de cette topologie est que l'augmentation du nombre de sauts d'une communication a une autre engendre une augmentation de la consommation d'énergie dans le réseau. Son avantage principal est qu'elle permet de maintenir la connectivité globale du réseau [136].

1.4.4.4 Topologie hybride

La topologie hybride [137] conjugue la topologie en étoile et la topologie maillée. Elle fournit un réseau robuste et minimise la consommation d'énergie des capteurs. Les nœuds ayant une capacité énergétique plus élevée assurent une communication multi saut. En revanche, ceux à faible puissance sont désactivés

1.4.5 Alimentation des WSNs

La source d'alimentation qui active les nœuds de capteurs dans WSN est l'un des composants des nœuds de capteurs. En raison du grand nombre de nœuds ou du déploiement des nœuds dans des endroits hostiles et difficiles d'accès, recharger ou remplacer les batteries est difficile et irréalisable pour la plupart des implémentations WSN. Les WSNs peuvent être classés en rechargeables ou non rechargeables en fonction du type de batterie utilisée par leurs nœuds. Une fois les batteries épuisées, les WSNs non rechargeables mourront. Cela se produit parce que chaque batterie a un budget énergétique limité. En revanche, les WSNs rechargeables (rWSN) utilisent des batteries qui peuvent être rechargées en utilisant des méthodes de transfert d'énergie éolienne, solaire et sans fil pour générer de l'énergie.

La composition chimique interne des batteries rechargeables détermine la capacité de récupérateur d'énergie. Le plomb scellé (SLA), le nickel-cadmium (NiCd), le nickel-hydrure métallique (NiMH), le lithium-ion, le lithium-polymère et d'autres technologies peuvent être rechargeables. Le tableau 1.6 suivant montre les spécifications des batteries rechargeables actuelles.

TABLE 1.6 – Comparaison entre les batteries rechargeables utilisées dans les WSN.

Type	Densité d'énergie (Wh/Kg)	Autodécharge (% : mois)	Efficacité (%)	Cycle de re- charge
Lead-acid	150-400	20 à 30	75-85	200 à 1000
Li-polymer	3000	<10	99.8	500 à 1000
Lithium-Ion	1800	<10	99.9	1200
Nickel-metal hy- dride	250-1000	20	66	1000
Nickel-cadmium	150	10	70-90	1500
Sealed-lead-acid	180	20	70-92	500-800

1.4.6 Défis des WSNs

Comme nous l'avons présenté dans la sous-section Domaines d'application de la PA, les WSNs ont prouvé leur efficacité pour réaliser et optimiser plusieurs tâches dans le processus agricole comme le contrôle de l'irrigation, la gestion des cultures, etc. Malheureusement, les caractéristiques et les capacités limitées des WSNs en termes d'énergie, routage des données, et interférence des radios, etc (Figure 1.13) font que les WSNs sont sujets à de nombreux défis. Beaucoup de recherches ont été conduites dans le but de pallier à ces limites. Notamment en ce qui concerne l'énergie [138], les WSNs nécessitent de l'énergie pour diverses opérations comme la collecte des données, la transmission des données et la surveillance des canaux de communication, etc. Les batteries fournissant de l'énergie doivent être changées ou rechargées après avoir été consommées. Il devient parfois difficile de recharger ou de changer les piles en raison des conditions démographiques. Le défi de recherche le plus crucial pour les chercheurs des WSNs est de concevoir, développer et mettre en œuvre des protocoles matériels et logiciels économes en énergie pour les WSNs [139].

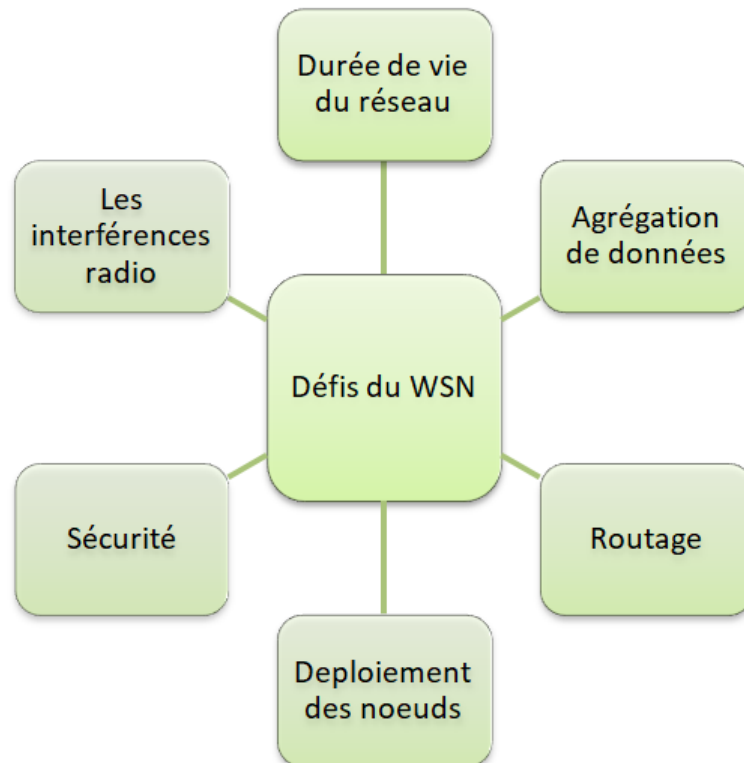


FIGURE 1.13 – Défis du WSN.

Un autre défi qui se renonce très importants pour les développeurs des WSNs est l'autogestion. Plusieurs nœuds capteurs peuvent être défaillants lors de l'utilisation du réseau. De ce fait, la topologie du réseau doit s'adapter avec ces changements pour maintenir la communication entre les différents éléments du réseau. Donc, une fois déployé, le WSN doit être capable de fonctionner de manière indépendante et doit être capable de configurer, d'adapter, de réparer et de maintenir le réseau [140].

De même pour le déploiement des nœuds qui consiste à trouver le meilleur emplacement des nœuds capteurs dans la région d'intérêt de manière à répondre le mieux aux critères de déploiement qui varient selon le type d'application [141]. Parmi ces critères on trouve la couverture du réseau, cette contrainte est parmi les problèmes les plus importants pour la réalisation d'un WSN. Elle est affectée essentiellement par le rayon de détection (R_d) du nœud capteur. D'autre part, la connectivité dépend de la portée de communication (rayon de communication : R_c) des nœuds capteurs et de leur disposition dans l'espace. De plus, il existe d'autres critères comme le coût de déploiement,

la sur-couverture, le routage de données qui doivent être pris en considération lors de l'emplacement des nœuds.

1.4.7 Majeurs objectifs de déploiement des WSNs

Les objectifs de déploiement des WSNs peuvent varier en fonction des applications spécifiques, mais ils incluent généralement les suivants :

1.4.7.1 Couverture

Pour définir le problème de couverture, on peut tout simplement dire que la couverture concerne la façon de s'assurer que chacun des points de la région à surveiller est couvert par les capteurs. Considérée comme l'une des métriques de la Qualité de Service (QoS), la couverture représente un facteur important lors du choix de déploiement d'un WSN, elle peut être complète ou partielle en fonction des besoins de l'application. Si n'importe quel point est surveillé par un seul nœud capteur on parle d'une couverture simple, et si le point est couvert par plusieurs nœuds capteurs, on parle dans ce cas-là de la sur-couverture [142]. Le problème de couverture dans WSN est essentiellement dû à trois raisons principales : pas assez de capteurs pour couvrir l'ensemble du retour sur investissement, portée de détection limitée et déploiement aléatoire. Étant donné que les capteurs fonctionnent avec une alimentation électrique limitée, certains d'entre eux pourraient s'éteindre, ce qui entraînerait des capteurs inadéquats pour couvrir entièrement la ROI entière, provoquant ainsi des trous. La plage de détection d'un capteur est limitée à un certain rayon, ce qui entraîne par conséquent un problème de couverture. Bien sûr, ce problème peut être résolu en utilisant un capteur avec une plage de détection plus large, mais ce type de capteur est plus cher [143]. La couverture dans les WSNs peut être classée en trois types :

i. Couverture de champ

Ce type permet d'avoir une couverture sur une zone en fonction des exigences de l'application. Une couverture complète est essentielle dans les applications qui exigent un haut degré de précision et elle est obtenue quand le taux de couverture est égal à 1. En fait, certaines applications ne nécessitent pas une couverture complète et une couverture partielle est acceptable, ce qui est un

moyen très efficace pour économiser l'énergie des capteurs et de prolonger la durée de vie du réseau puisque le nombre de capteurs déployés dans la zone efficace est inférieur à celui requis pour une couverture complète [124], qui est illustré sur la Figure 1.14.

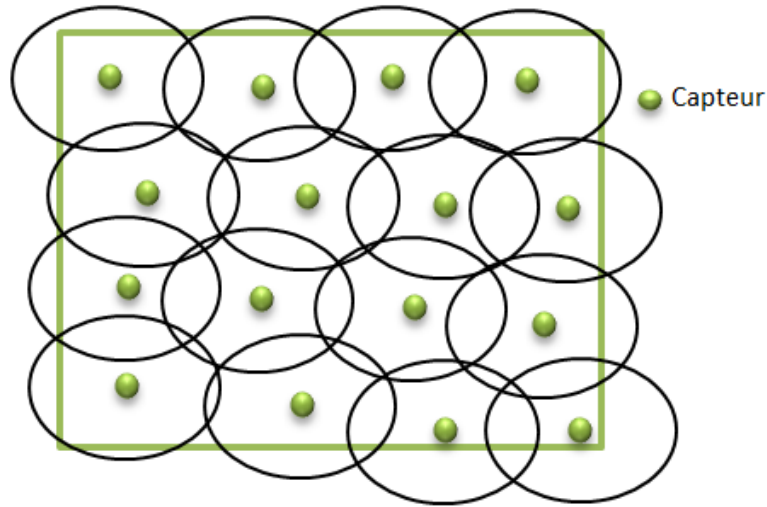


FIGURE 1.14 – Modèle de couverture de champ.

ii. Couverture de cible

Dans de nombreuses applications, il suffit de surveiller quelques points d'intérêts spécifiques à la zone d'application et il n'est pas intéressant de recouvrir toute la zone. Dans ce cas, on a recours à une couverture de cible. En fait, le nombre de cibles est généralement fixe et il est donc clair que le coût de déploiement du réseau se voit réduit puisqu'il s'agit d'une couverture partielle [124], cela est montré sur la Figure 1.15.

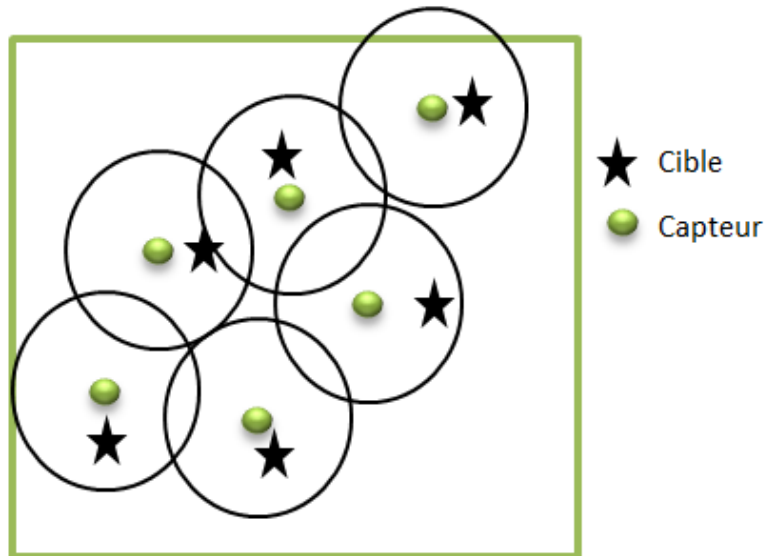


FIGURE 1.15 – Modèle de couverture des points d'intérêts (cibles).

iii. Couverture de barrière

Dans la couverture par barrière, les nœuds sont déployés de manière à former une barrière sur un chemin spécifique (Figure 1.16) et transmettent l'information s'ils détectent des activités possibles d'un intrus pour traverser la barrière. L'objectif principal est que chaque point de la zone d'intérêt soit dans le champ de couverture d'au moins un capteur[124].

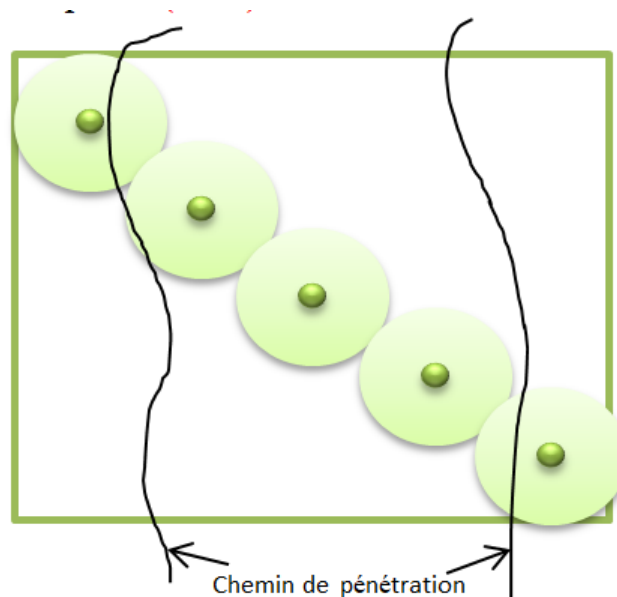


FIGURE 1.16 – Modèle de couverture de barrières.

1.4.7.2 Connectivité

Si dans un WSN, deux nœuds capteurs peuvent communiquer on peut dire que ces deux nœuds sont connectés. La communication peut être faite à un seul saut (communication directe), à multiples sauts (communication indirecte). Dans le WSN, le réseau est considéré comme connecté s'il existe au moins un chemin entre les nœuds puits et chaque nœud capteur dans la zone considérée [144]. Afin de bien surveiller une région donnée, il est insuffisant d'assurer la couverture du réseau seulement, il faut aussi tenir compte de la connectivité. Car les nœuds capteurs doivent être capables de signaler immédiatement aux nœuds puits tout événement détecté. La connectivité réseau complète peut également être simple (1-connectivité) : s'il y a un seul chemin depuis tout nœud capteur jusqu'aux nœuds puits, ou multiple (k-connectivité) : si plusieurs chemins disjoints existent entre un nœud quelconque et les nœuds puits. Il existe plusieurs modèles de propagation radio qui peuvent exprimer le comportement de l'onde transmise par un nœud dans un environnement précis. Ces modèles sont présentés dans les parties suivantes :

i. Modèle de FRIS

Ce modèle est l'un des premiers modèles utilisés, il exprime un rapport de puissance du signal reçu et celui émis par deux antennes isotropes séparées d'une distance R . Le rapport est donné par l'équation suivante [145] :

$$\frac{P_r}{P_t} = n \times \left[\left(\frac{\lambda}{4\pi R} \right)^2 \times G_t \times G_r \right] \quad (1.1)$$

Où P_r et P_t représentent la puissance de récupération et la puissance de transmission.

G_t et G_r représentent respectivement le gain de transmission et le gain de réception de l'antenne. Le terme $(\frac{\lambda}{4\pi R})^2$ est appelé et facteur de perte en espace libre avec λ la longueur d'onde et R la distance entre les deux antennes. Enfin, n représente les différentes pertes liées à la désadaptation (réflexion et polarisation).

ii. Modèle Multi-Wall (MWM)

Un modèle plus sophistiqué a été développé par Motley et Keenan. Ce modèle

prend en compte tous les murs et planchers pénétrant par des pertes de pénétration individuelles en fonction de leur épaisseur et de leur matériel. Ce modèle de propagation est le plus approprié aux environnements intérieurs (indoor) [146]. Il prend en compte les atténuations dues à la pénétration de l'onde entre l'émetteur et le récepteur dans les obstacles. Le modèle est représenté par l'équation suivante :

$$PL_{MWF}[dB] = FSL + 10 \times n \times \log_{10}(d) + \sum_{i=1}^I \sum_{k=1}^{k_{wi}} (L_{fik}) + \sum_{j=1}^J \sum_{k=1}^{k_{wj}} (L_{fjk}) \quad (1.2)$$

$$RSSI[dBm] = P_{rx}[dBm] + PL_{MWF} \quad (1.3)$$

Ou :

- FSL : conception des Pertes en Espace Libre ;
- L_{wik} : est l'atténuation du kème mur de la catégorie i ;
- L_{fjk} : est l'atténuation du kème étage de la catégorie j ;
- K_{wi} : est le nombre des murs de la catégorie i ;
- K_{fj} : est le nombre d'étages de la catégorie j.

1.4.7.3 Durée de vie

La durée de vie d'un réseau est principalement influencée par la quantité d'énergie qu'il utilise. En raison du fait que les capteurs et les relais sont principalement alimentés par des batteries, leur durée de vie est limitée. Les composants radiofréquences consomment plus d'énergie que l'unité de traitement des données. Ainsi, la consommation d'énergie de l'unité de communication, qui dépend de la distance entre les nœuds et du nombre de paquets transmis, déterminera la durée de vie. Il sera préférable d'utiliser une topologie en étoile avec une communication directe entre les capteurs et les routeurs afin d'éviter les phénomènes de sur écoute [147].

1.4.8 Domaines d'utilisation des WSN

Les domaines d'applications des WSNs sont de plus en plus nombreux grâce à leurs petites tailles et leurs capacités de surveillance d'un bon nombre de phénomènes

physiques, ainsi que l'élargissement des gammes de capteurs disponibles (mouvement, température, ...) et l'évolution des supports de communication sans fil. Par conséquent les WSNs ont montré de bons résultats dans les domaines : militaires, médicaux, environnementaux, commerciaux, etc. Comme le présente la Figure 1.17.

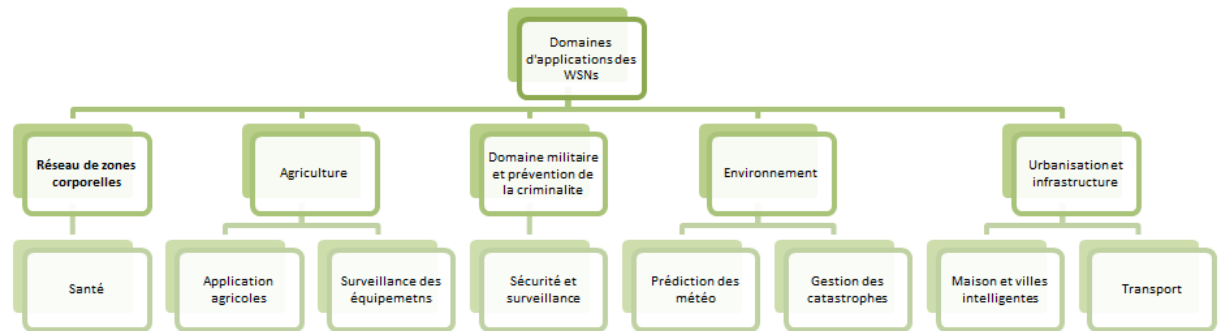


FIGURE 1.17 – Taxonomie des applications des WSNs.

1.4.8.1 Applications militaire et prévention de la criminalité

Un WSN peut être déployé dans des secteurs stratégiques ou difficiles d'accès, afin d'assurer des applications de surveillance (Figure 1.18), reconnaissance et ciblage, fournir des informations en temps réel sur les conditions du terrain et analyser le champ de bataille avant d'y envoyer du renfort [148].



FIGURE 1.18 – Surveillance militaire par WSN [149].

1.4.8.2 Applications en réseau de zones corporelles

Les WSNs sont utilisés dans le domaine de la télémédecine et d'aide aux patients (Figure 1.19). Ces réseaux peuvent surveiller les signes vitaux des patients en temps réel, permettant une gestion plus proactive des soins de santé. Par exemple, il existe aussi des gélules multi-capteurs pouvant être avalées qui permettent, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur du corps humain [41]. De plus, ils peuvent surveiller les conditions environnementales dans les hôpitaux pour assurer un environnement sûr et hygiénique.

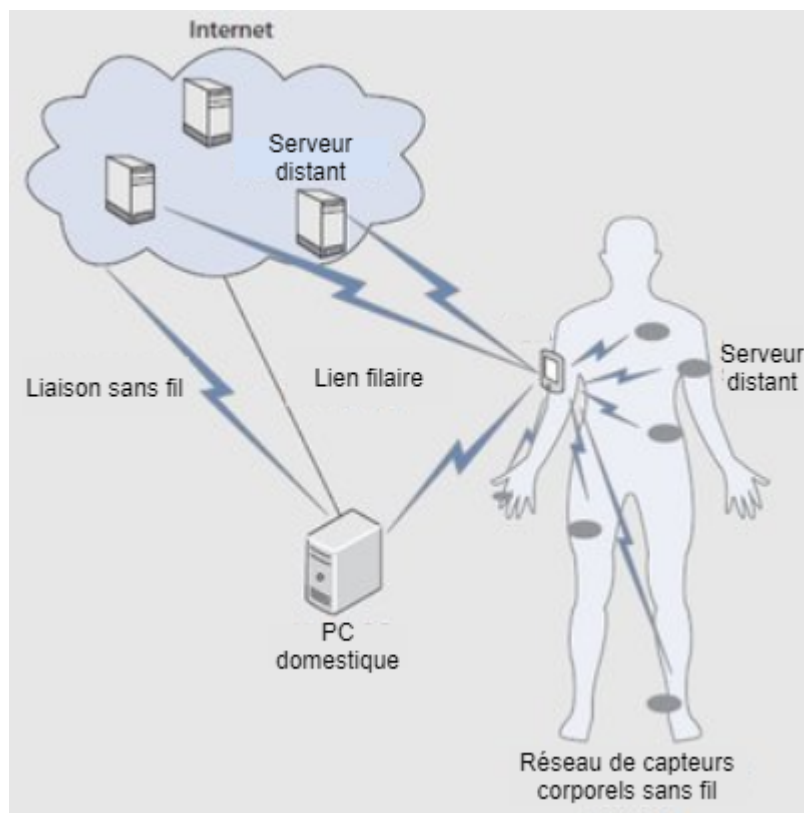


FIGURE 1.19 – Télémédecine par WSN.

1.4.8.3 Applications en agriculture

les WSNs offrent de nombreuses améliorations au processus agricole grâce à leurs diverses applications. Ils sont utilisés pour la surveillance des cultures, permettant de suivre en temps réel l'humidité du sol, la température, la luminosité et d'autres conditions environnementales cruciales. Cette surveillance aide à optimiser l'irrigation et l'application de fertilisants, garantissant des rendements maximaux.

La détection des maladies est une autre application cruciale des WSNs. Les capteurs peuvent identifier les premiers signes de maladies des plantes en surveillant les changements dans les conditions des cultures, ce qui permet aux agriculteurs de prendre des mesures préventives avant que les maladies ne se propagent.

les WSNs sont également utilisés pour la surveillance du bétail, permettant de suivre les mouvements et la santé des animaux en temps réel. Cela inclut la détection de comportements anormaux qui pourraient indiquer des problèmes de santé, ainsi que la localisation des animaux pour éviter les pertes et optimiser les parcours de pâturage.

Les systèmes d'irrigation intelligents bénéficient également des WSNs en ajustant automatiquement l'irrigation en fonction des données collectées par les capteurs. Cela permet une utilisation plus efficace de l'eau, réduisant ainsi le gaspillage et assurant que les plantes reçoivent l'eau nécessaire à leur croissance optimale.

les WSNs jouent un rôle clé dans la détection des incidents naturels tels que les incendies et les inondations. Des capteurs de température et d'humidité peuvent être dispersés par avion ou installés sur le terrain pour détecter des anomalies environnementales. Ces capteurs fournissent des alertes précoces, permettant une intervention rapide pour minimiser les dégâts causés par ces catastrophes naturelles [41].

L'étude de l'empreinte carbone (Figure 1.20) est une autre application des WSN. En surveillant les émissions de gaz à effet de serre et d'autres indicateurs environnementaux, les agriculteurs peuvent évaluer et réduire l'empreinte carbone de leurs activités agricoles, contribuant ainsi à des pratiques plus durables.



FIGURE 1.20 – Systèmes de surveillance du sol par WSN [150].

1.4.8.4 Application dans l'environnement

La prévision météorologique avec WSN peut être essentielle pour sauver de nombreuses vies des catastrophes environnementales telles que les inondations, les tremblements de terre, les glissements de terrain, les incendies de forêt, etc. En raison de changements récurrents, la prévision météorologique constitue un défi majeur pour le département de météorologie (Figure 1.21). Les capteurs utilisés doivent surveiller de nombreuses activités extérieures telles que le refroidissement éolien, les précipitations et la neige, la température, etc. Ces ensembles de données météorologiques sont ensuite utilisés pour prédire la météo à l'aide de techniques d'exploration de données dont le processus comporte trois étapes : prétraitement des données, formation du modèle, puis prédiction [151].

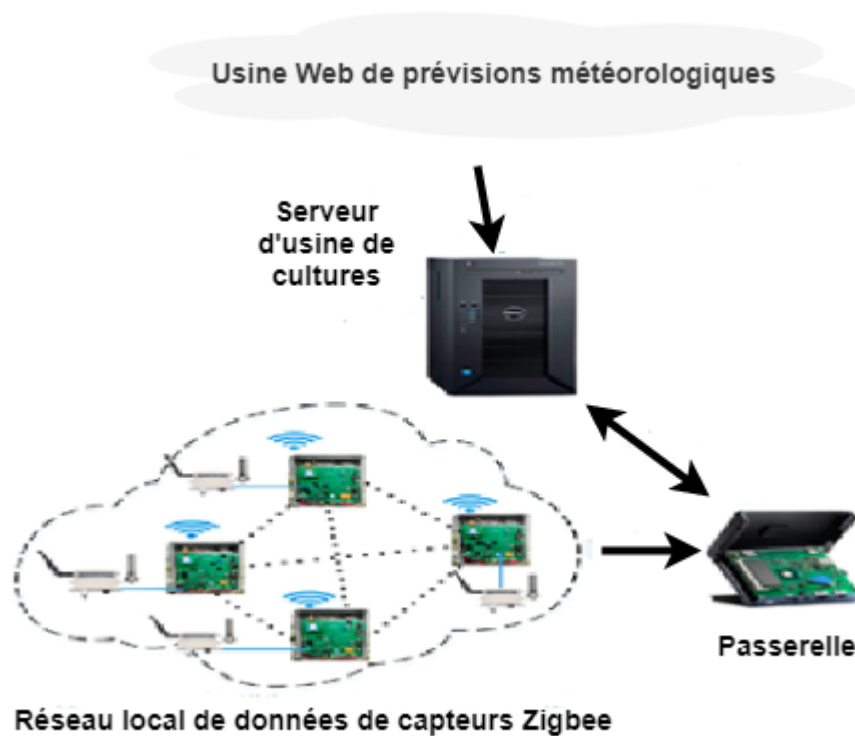


FIGURE 1.21 – Application du WSN pour la prédiction de la météo.

1.4.8.5 Applications en urbanisation et infrastructure

Les WSNs peuvent donner des informations sur l'environnement du bâtiment, ainsi sur les phénomènes physiques qui se produisent dans son enceinte. Ce réseau peut aussi être un outil de protection et de surveillance vu leur coût et leur taille qui ne cessent de diminuer, Ce qui explique pourquoi ils sont, en général, les équipements les plus privilégiés et couramment utilisés comme infrastructure réseau dans les habitats intelligents [152, 153] comme le fameux bâtiment the Edge (Figure 1.22).



FIGURE 1.22 – Façade du bâtiment The Edge [154].

1.5 Techniques de déploiement des WSNs

En raison des contraintes d'énergie, de traitement et de communication, la conception des WSNs est un problème complexe. L'un des aspects les plus critiques de cette conception est le déploiement des nœuds dans la région d'intérêt (RoI), car il influence la connectivité, la couverture, la qualité de service et la durée de vie du réseau. Plusieurs stratégies de déploiement sont utilisées, classées en méthodes statiques ou dynamiques.

Les méthodes statiques incluent les stratégies aléatoires, où les nœuds sont distribués de manière aléatoire [155], et les méthodes déterministes, qui placent les nœuds selon des contraintes spécifiques pour atteindre des objectifs prédéfinis [156]. Les méthodes dynamiques comprennent le déploiement centralisé, où un cluster-head coordonne le déploiement, et le déploiement distribué, où chaque capteur utilise des informations locales pour ajuster sa position en fonction des capteurs voisins et des obstacles.

Dans la littérature, les algorithmes de déploiement sont regroupés en quatre catégories principales selon leur approche mathématique (Figure 1.23) : les algorithmes génétiques (GA), la géométrie computationnelle (CG), les champs de potentiel artificiels (APF) et l'optimisation par essaims de particules (PSO). Les GA visent à optimiser la conception des WSNs avec plusieurs objectifs [157], tandis que les algorithmes basés sur CG comme le diagramme de Voronoi et la triangulation de Delaunay sont utilisés pour évaluer la couverture et détecter les zones non couvertes [158]. Les APF sont

utilisés principalement pour les WSNs mobiles afin d'éviter les obstacles [159], tandis que le PSO s'inspire du comportement collectif pour optimiser le déploiement des capteurs mobiles [160]. Les catégories d'algorithmes peuvent être appliquées dans divers scénarios de déploiement, statique ou dynamique. Les approches GA et PSO sont particulièrement adaptées pour le déploiement multi-objectif, permettant de déterminer efficacement les fonctions objectif nécessaires [156]. Les GA nécessitent moins de paramètres d'optimisation par rapport au PSO, produisant non seulement une solution unique mais aussi un ensemble de solutions permettant d'approximer l'ensemble du front de Pareto. Pour notre étude, nous avons choisi l'application des variantes des MOEAs pour trouver le déploiement optimal des WSNs qui répond aux contraintes spécifiques de notre modèle proposé.

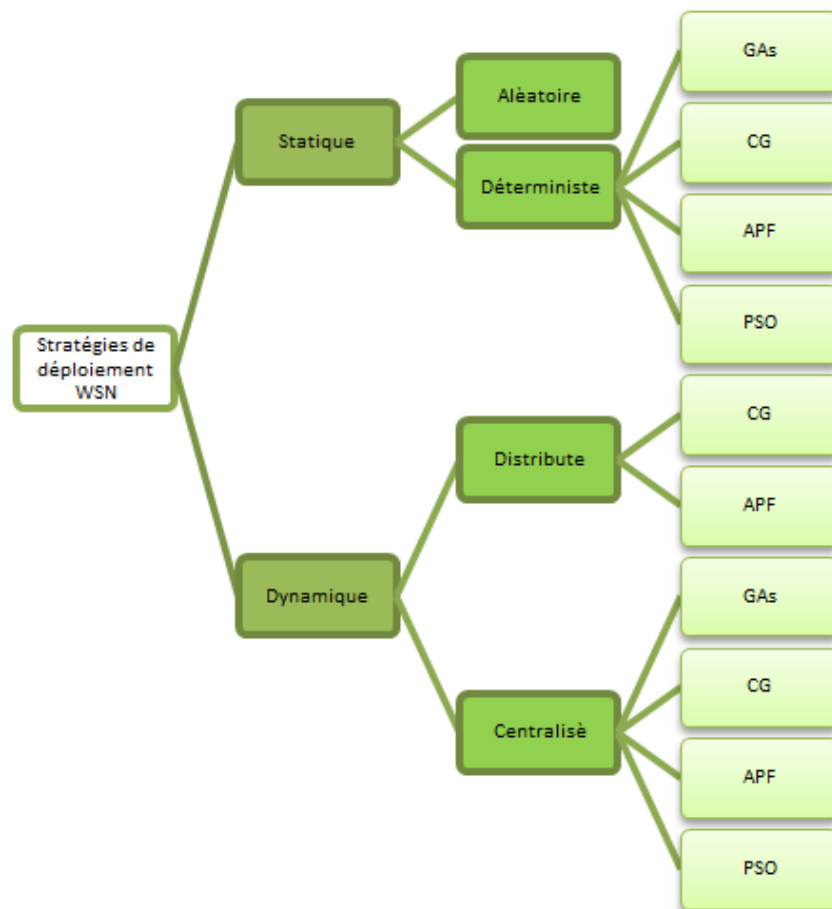


FIGURE 1.23 – Méthodes de déploiement des nœuds dans un WSN.

Le déploiement statique des nœuds capteurs dans une zone d'intérêt peut

généralement être de manière planifiée ou aléatoire. Le type de nœuds capteurs, l'application et l'environnement dans lequel les nœuds fonctionnent sont tous des facteurs qui influencent le choix du type de déploiement. Dans des situations où nous n'avons pas une connaissance préalable des champs comme sur un champ de bataille, le déploiement aléatoire peut être le seul choix. Cependant, le déploiement déterministe est souvent nécessaire lorsque les nœuds capteurs coûtent cher ou lorsque leur position a un impact assez important sur leur fonctionnement.

1.5.1 Déploiement statique avec des méthodes exactes

Ce type de déploiement est utilisé lorsque l'environnement est accessible ou connu, donc il est possible de placer précisément les nœuds sur la zone dans des positions fixes et connues selon un plan prédéfini. C'est ainsi, par exemple, que sont mis en place les capteurs chargés de réguler la climatisation d'un immeuble ou de surveiller les constantes médicales des malades à distance. On trouve dans ce type de déploiement plusieurs algorithmes qui peuvent être utilisés comme la méthode de gradient, de simplexe, de Branch et Bound et le diagramme de Voronoi. Dans la partie suivante, nous allons donner une brève description des différentes méthodes[161].

1.5.1.1 Méthode de gradient

La méthode de gradient est parmi les méthodes les plus anciennes qui permettent la résolution des problèmes non-linéaires et l'optimisation des fonctions réelles, elle se base sur la connaissance de la dérivée de la fonction objective en tous points de l'espace de recherche. Cette méthode ajuste itérativement les positions des capteurs en fonction d'un gradient calculé par rapport à une fonction objective, comme la densité de couverture attendue ou la qualité du signal reçu[143].

1.5.1.2 Méthode du simplexe

Développée par John A. Nelder et Roger Mead, cette méthode fait partie des approches locales d'optimisation les plus répandues. Sa particularité réside dans le fait qu'au lieu d'améliorer une seule solution à chaque itération, l'algorithme agit sur un ensemble de solutions appelé polytope. Il explore de manière systématique l'espace de

recherche multidimensionnel pour identifier la solution optimale. Cette méthode est principalement utilisée pour résoudre des problèmes d'optimisation linéaire [162].

1.5.1.3 Branch et Bound

Cette méthode divise l'espace de recherche en sous-problèmes plus petits (branchement) et évalue systématiquement chaque branche, en utilisant des bornes (Bound) pour éliminer les solutions non prometteuses (Figure 1.24). Si les sous-problèmes sont aussi difficiles à résoudre, ils seront divisés aussi comme montre la figure suivante. Il existe une variante plus utilisée de cet algorithme qui s'appelle Branch cut [163].

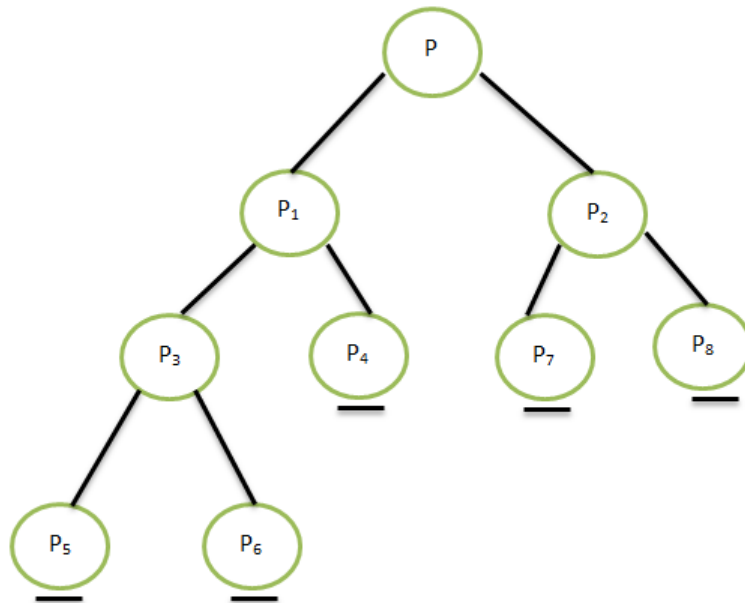


FIGURE 1.24 – La division du problème P en plusieurs sous-problèmes par l'algorithme Branch Bound.

Les méthodes déterministes restent toujours les moins performantes quand il s'agit des problèmes de grande taille. L'inconvénient majeur de ce type de méthode est le temps de calcul qui augmente exponentiellement en fonction de la complexité et de la taille du problème.

D'où semble logique d'essayer de résoudre ces problèmes avec des méthodes d'optimisation non-déterministe aussi appelées stochastiques qui trouvent des solutions s'approchant de l'optimale dans des délais plus courts.

1.5.1.4 Méthode de diagramme de Voronoi

Le diagramme de Voronoi, une méthode courante en géométrie computationnelle pour le déploiement des WSN, divise l'espace autour de chaque capteur (ou site) en zones distinctes. Chaque point à l'intérieur d'une zone de Voronoi est plus proche de son capteur associé que de tout autre capteur du réseau. Dans le déploiement déterministe des WSN, ce principe est utilisé de la manière suivante : Chaque capteur est considéré comme un site dans le diagramme de Voronoi, et les zones de Voronoi définissent les régions où les signaux captés par chaque capteur sont les plus forts. Pour assurer une couverture maximale de la zone d'intérêt sans chevauchement excessif, les capteurs sont positionnés de sorte que chaque région de Voronoi soit couverte par au moins un capteur. De plus, la structure du diagramme de Voronoi favorise une meilleure connectivité entre les capteurs en minimisant les distances entre eux. Les capteurs voisins, dont les zones de Voronoi se touchent, bénéficient d'une communication plus efficace et réduisent les zones non couvertes. Le diagramme de Voronoi est largement appliqué dans des domaines tels que la surveillance environnementale, la détection des intrusions, et d'autres applications où une couverture étendue et une connectivité robuste sont cruciales pour le bon fonctionnement du WSN[164].

1.5.2 Déploiement statique avec des méthodes heuristiques

Si la zone d'intérêt présente des obstacles d'accès ou elle est considérée comme sensible, le déploiement déterministe des nœuds capteurs devient impraticable, ce qui justifie l'utilisation d'une méthode aléatoire. Cette approche consiste à disperser les nœuds capteurs à l'aide d'avions ou de drones, ce qui entraîne une distribution aléatoire des capteurs sur le terrain. Toutefois, cette méthode peut être complexe étant donné la fragilité des capteurs existants, et elle peut aboutir à des performances sous-optimales, notamment en termes de couverture et de connectivité du réseau. Les méthodes d'optimisation stochastique (Stochastic Optimization, SO) jouent un rôle crucial dans l'amélioration de ces déploiements aléatoires. Les méta-heuristiques, par exemple, sont des techniques non déterministes bien établies qui exploitent des concepts probabilistes et des mécanismes aléatoires pour résoudre des problèmes d'optimisation difficiles où aucune méthode classique ne s'avère efficace. Ces algorithmes

itératifs visent à converger vers un optimum global en explorant de manière itérative l'espace des solutions, apprenant ainsi les caractéristiques du problème pour trouver une approximation de la meilleure solution possible. Ils englobent une variété de techniques, depuis la recherche locale jusqu'aux stratégies complexes de recherche globale.

Par conséquent, l'utilisation de méthodes du SO permet d'améliorer significativement les performances des déploiements aléatoires des WSN, en adaptant dynamiquement les configurations des capteurs aux conditions spécifiques de chaque environnement tout en maximisant la couverture et la connectivité du réseau. Ces avancées sont cruciales pour répondre aux exigences variées des applications telles que la surveillance environnementale, la gestion des catastrophes et d'autres domaines où une collecte de données fiable et efficace est essentielle[165]. Dans cette partie, nous allons expliquer les méthodes les plus utilisées en déploiement statique avec des méthodes heuristiques des WSN.

1.5.2.1 Recherche Tabou

Proposée par Fred Glover, l'idée de cette méthode principale consiste à garder en mémoire l'historique des solutions ou des caractéristiques des solutions déjà explorées. Elle consiste en la création d'une liste qui contient toutes les solutions récemment explorées qui deviendront tabous. Ainsi, l'algorithme ne pourra plus les choisir comme solution. Ceci permet à l'algorithme de ne pas retomber dans l'optimum local autorisé il vient de sortir. Ainsi, à chaque itération, l'algorithme choisit le meilleur voisin non tabou même si celui-ci détériore la valeur de la fonction objective. Ceci permet une meilleure exploration de l'espace de recherche.

Dans le contexte du déploiement aléatoire des WSN, la recherche tabou pourrait être utilisée pour guider la répartition des capteurs de manière à optimiser la couverture du territoire tout en évitant les configurations qui pourraient conduire à des lacunes de couverture ou à une connectivité insuffisante entre les capteurs. Par exemple, elle pourrait aider à déterminer les meilleurs emplacements pour les capteurs en tenant compte des contraintes environnementales et des exigences de performance du réseau[166].

1.5.2.2 Intelligence en Essaim

L'optimisation par essaims est une méta-heuristique d'optimisation, inventée par Russel Eberhart et James Kennedy en 1995. Ces algorithmes s'appuient sur le concept d'auto-organisation ou un groupe d'individus peu intelligents peuvent posséder une organisation globale complexe. Les algorithmes de colonies de fourmis sont les méthodes d'intelligence en essaim les plus utilisées. Cette méthode a montré son efficacité dans l'optimisation du déploiement aléatoire des WSNs sur différents aspects[167].

- Optimisation de la couverture : Le PSO peut être utilisé pour rechercher de manière efficace les positions optimales des nœuds capteurs dans une zone donnée. En utilisant des mécanismes de mise à jour itérative basés sur des particules (capteurs), le PSO peut converger vers des solutions qui maximisent la couverture de la zone d'intérêt tout en minimisant les zones non couvertes.
- Gestion de la connectivité : Assurer une connectivité efficace entre les nœuds capteurs est essentiel dans les WSN. Le PSO peut être configuré pour optimiser non seulement la couverture mais aussi la qualité de la connectivité entre les capteurs, en ajustant les positions des nœuds pour maximiser la transmission des données et minimiser les retards de communication.
- Considérations sur l'énergie : Le PSO peut être adapté pour optimiser la consommation d'énergie des nœuds capteurs. En minimisant les distances de communication et en optimisant les itinéraires de transmission, il contribue à prolonger la durée de vie du réseau en utilisant efficacement les ressources énergétiques limitées des capteurs.

1.5.2.3 Algorithmes d'optimisation multi-objectifs

Lorsque la zone d'intérêt présente des défis d'accès ou de sensibilité, le recours à un déploiement aléatoire des WSNs devient une option privilégiée. Cette méthode consiste à disperser les nœuds capteurs à l'aide d'aéronefs tels que des avions ou des drones, générant ainsi une répartition aléatoire des nœuds. Cette approche, bien que pratique pour couvrir rapidement de vastes zones, présente des défis importants. Les capteurs étant souvent fragiles, ce type de déploiement peut conduire à des performances sous-optimales, notamment en termes de couverture et de connectivité. Par exemple, il

ne garantit pas nécessairement que chaque partie de la zone d'intérêt soit surveillée de manière optimale ni que tous les capteurs soient connectés de manière efficace pour transmettre les données collectées. Les méthodes du SO, telles que les méta-heuristiques, sont couramment employées dans le contexte du déploiement aléatoire des WSN. Ces approches non déterministes tirent parti de concepts probabilistes et de mécanismes aléatoires pour résoudre des problèmes d'optimisation complexes où les méthodes classiques ne sont pas efficaces. Les méta-heuristiques, comme les GAs et les PSOs, sont particulièrement adaptées. Elles itèrent à travers des solutions potentielles, tentant d'apprendre et d'améliorer les caractéristiques du problème pour trouver une solution proche de l'optimum global.

En pratique, les méta-heuristiques permettent d'explorer de manière efficace l'espace des solutions possibles dans un déploiement aléatoire des WSN. Elles ajustent les positions des capteurs pour maximiser la couverture de la zone tout en maintenant une connectivité adéquate entre les nœuds. Cela est crucial dans des domaines tels que la surveillance environnementale, la gestion des ressources naturelles et la sécurité, où une couverture complète et une collecte de données robuste sont essentielles[168].

1.5.3 Déploiement statique aléatoire

Le déploiement statique aléatoire est une stratégie dans laquelle les nœuds capteurs sont dispersés de manière aléatoire sur la zone à surveiller, sans ajustement ultérieur de leur position. Cette méthode est particulièrement utilisée lorsque le terrain est difficile d'accès ou lorsque le déploiement manuel est impraticable, comme par exemple par largage aérien. Bien qu'elle soit simple et rapide à mettre en œuvre, cette approche peut entraîner une couverture inégale ou des zones non connectées, ce qui nécessite parfois des mécanismes de redondance ou d'optimisation en aval.

1.6 Conclusion

En conclusion, l'intégration des WSNs dans la PA représente une avancée significative vers une gestion agricole plus efficace, durable et respectueuse de l'environnement. les WSNs permettent une surveillance continue et détaillée des paramètres essentiels

du sol et des cultures, fournissant aux agriculteurs les données nécessaires pour optimiser les interventions et maximiser les rendements. Cependant, le déploiement optimal de ces capteurs reste un défi majeur, particulièrement dans des environnements difficiles d'accès ou vastes.

Les algorithmes d'optimisation multi-objectifs jouent un rôle crucial dans la résolution de ce problème. Contrairement aux approches traditionnelles qui se concentrent sur un seul objectif, ces algorithmes permettent de considérer simultanément plusieurs critères importants, tels que la couverture, la connectivité, la consommation d'énergie et les coûts. Les algorithmes méta-heuristiques, tels que les GAs et les PSOs, sont particulièrement efficaces pour trouver des solutions équilibrées qui satisfont ces multiples objectifs.

Ces techniques avancées d'optimisation permettent non seulement de maximiser la couverture et la fiabilité des WSN, mais aussi de prolonger la durée de vie des capteurs en minimisant leur consommation d'énergie. Elles facilitent également l'adaptation des réseaux aux conditions changeantes du terrain et des cultures, assurant ainsi une collecte de données continue et précise. En intégrant ces algorithmes dans les stratégies de déploiement des WSN, les agriculteurs peuvent obtenir une vision plus complète et nuancée de leurs champs, permettant des interventions plus ciblées et efficaces.

Ainsi, l'utilisation des algorithmes d'optimisation multi-objectifs dans le déploiement des WSNs constitue une avancée déterminante pour la PA. Elle ouvre la voie à des pratiques agricoles plus intelligentes et durables, capables de répondre aux défis croissants de la sécurité alimentaire et de la conservation des ressources. Les recherches futures devront continuer à affiner ces algorithmes et à explorer de nouvelles approches pour optimiser encore davantage le déploiement et l'utilisation des WSN, assurant ainsi une PA toujours plus performante et résiliente.

Chapitre 2

Application des MOEAs pour l'optimisation du déploiement des nœuds

2.1 introduction

Les problèmes d'optimisation multi-objectifs (MOPs) sont omniprésents dans diverses disciplines, allant de l'ingénierie à la gestion environnementale. Ces problèmes impliquent la maximisation ou la minimisation simultanée de plusieurs objectifs souvent conflictuels. Pour résoudre ces MOPs, les algorithmes évolutionnaires multi-objectifs (MOEAs) se sont révélés particulièrement efficaces. Les MOEAs, tels que SPEA II (Strength Pareto Evolutionary Algorithm) et NSGA II (Non-dominated Sorting Genetic Algorithm), permettent d'obtenir une approximation de l'ensemble de Pareto, offrant ainsi une variété de solutions optimales selon des critères multiples.

Dans le contexte de l'agriculture, le déploiement optimal d'un WSN est un problème typique de MOP. Les objectifs incluent généralement la couverture maximale de la parcelle, la connectivité entre les nœuds, la minimisation des coûts de déploiement et la durabilité du réseau. Adapter les MOEAs à ce problème spécifique présente des défis uniques mais également des opportunités significatives pour améliorer l'efficacité et la rentabilité des opérations agricoles.

Ce chapitre explore l'adaptation des MOEAs pour le déploiement optimal des WSNs en agriculture. Une attention particulière est portée sur la comparaison entre deux algorithmes populaires : SPEA II et NSGA II. Les performances de ces algorithmes sont évaluées en termes de convergence vers l'optimalité et de diversité des solutions proposées. Enfin, les limites du modèle proposé sont discutées pour identifier les axes d'amélioration et les défis futurs.

2.2 Optimisation des problèmes multi-objectifs

De nombreux domaines industriels comme l'ingénierie, la conception, la bio-informatique, la logistique, les transports, les télécommunications, l'environnement, l'aéronautique, la finance et problèmes d'optimisation du monde réel sont concernés par des problèmes d'optimisation vastes et complexes impliquant de nombreux critères[169]. En effet, les MOPs sont décrits à l'aide de plusieurs critères souvent contradictoires et nécessitent une optimisation simultanée. À la différence des

problèmes mono-objectifs où l'optimum recherché est clairement défini, la solution optimale d'un MOP reste à décrire de manière formelle [170].

Effectivement, lorsqu'il s'agit d'un problème avec deux objectifs contradictoires, la solution optimale recherchée consiste en un ensemble de points qui correspond aux meilleurs compromis envisageables pour résoudre sa problématique. Par exemple, dans la conception d'un produit donné, il faut minimiser à la fois ses coûts et son impact environnemental et maximiser sa qualité. La complexité des MOPs devient de plus en plus importante par rapport à la taille du problème à résoudre et de la complexité du problème lui-même (par exemple, nombre d'objectifs, taille de l'espace de recherche). Les MOPs peuvent être regroupés dans des groupes plus ou moins nombreux selon les caractéristiques de leurs éléments. Cette classification permet par la suite la conception de la méthode d'optimisation [171]. En générale, les MOPs sont classés comme suivant [172] :

- Les contraintes liées aux variables. Un problème d'optimisation peut ne pas comporter de contraintes sur les variables de décision, ou bien avoir des contraintes d'égalité ou d'inégalité.
- La nature de l'espace de décision. Si certaines variables ne peuvent prendre que des valeurs entières, un problème est appelé un problème d'optimisation multi-objectif entière. Si toutes les variables peuvent prendre des valeurs réelles, le problème est connu sous le nom d'optimisation multi objectif continu.
- Forme des fonctions but et espace possible. Un MOP est convexe lorsque l'ensemble des fonctions objectifs et de l'espace des variables réalisables est convexe.
- Type de fonctions objectives et de contraintes. le MOP est linéaire lorsque toutes les fonctions objectives et contraintes sont linéaires. Si une contrainte ou un objectif n'est pas linéaire, le problème est appelé MOP non linéaire.

La solution optimale pour les MOPs n'est pas une solution unique comme pour les problèmes d'optimisation mono-objective, mais un ensemble de solutions définies comme des solutions optimales de Pareto. Une solution est Pareto optimale s'il n'est pas possible d'améliorer un objectif donné sans détériorer au moins un autre objectif. Cet ensemble de solutions représente les solutions de compromis entre les différents

objectifs en conflit. Le but principal de la résolution d'un MOP est d'obtenir l'ensemble optimal de Pareto, et par conséquent, le front de Pareto [173].

Par rapport à l'optimisation mono-objective, la difficulté à résoudre les MOPs réside dans les faits généraux suivants [174] :

1. Le nombre de solutions optimales de Pareto augmente avec la taille du problème et principalement avec le nombre d'objectifs considérés. En effet, au moins toutes les solutions de Pareto d'un problème à n objectifs sont des solutions de Pareto nécessaires du même problème avec une fonction objectif supplémentaire.
2. La structure du Front de Pareto dépend du MOP étudié. Par exemple, les solutions optimales de Pareto peuvent être localisées sur la frontière et à l'intérieur de l'enveloppe convexe des solutions réalisables.
3. Le nombre de solutions optimales de Pareto augmente avec la taille du problème et principalement avec le nombre d'objectifs considérés. En effet, au moins toutes les solutions de Pareto d'un problème à n objectifs sont des solutions de Pareto nécessaires du même problème avec une fonction objectif supplémentaire.
4. La structure du Front de Pareto dépend du MOP étudié. Par exemple, les solutions optimales de Pareto peuvent être localisées sur la frontière et à l'intérieur de l'enveloppe convexe des solutions réalisables.

2.2.1 Paramètres de l'optimisation multi-objective

2.2.1.1 Dominance au sens de Pareto

L'idée d'utiliser la dominance au sens de Pareto a été proposée par Goldberg pour résoudre les problèmes proposés par Schaffer. L'auteur suggère d'utiliser le concept d'optimalité de Pareto pour respecter l'intégralité de chaque critère au lieu de comparer a priori les valeurs de différents critères. L'utilisation d'une sélection basée sur la notion de dominance de Pareto entraîne la convergence de la population vers un

ensemble de solutions efficaces. Ce concept ne permet pas de choisir une alternative plutôt qu'une autre mais il apporte une aide précieuse au décideur.

Dans un problème de minimisation, un vecteur de décision $a \in X$ domine un autre vecteur de décision $b \in X$ et on écrit $a < b$ [175], si et seulement si :

1. La solution a n'est pas pire que b dans tous les objectifs.
2. a est meilleure que b dans au moins un seul objectif.

Si la relation 1 est vérifiée alors que la relation 2 n'est pas vérifiée, dans ce cas, on parle de **dominance faible**.

2.2.1.2 Pareto optimal

Soit $x = [x_1, \dots, x_n]$ un vecteur de décision avec : x (L'espace réalisable), x est dite Pareto optimal, s'il n'existe pas une solution y qui domine x . Une solution Pareto optimale est appelée aussi : solution efficace, non inférieure ou non dominée solution [176].

2.2.1.3 Front de Pareto

Le front (frontière) de Pareto est l'ensemble des solutions Pareto optimales qui sont composées des points, ne sont dominés par aucun autre le front de Pareto appelé aussi surface de compromise ou l'ensemble des solutions efficaces. La Figure 2.1, représente le Front du Pareto pour un problème de minimisation et maximisation de deux fonctions objective [1].

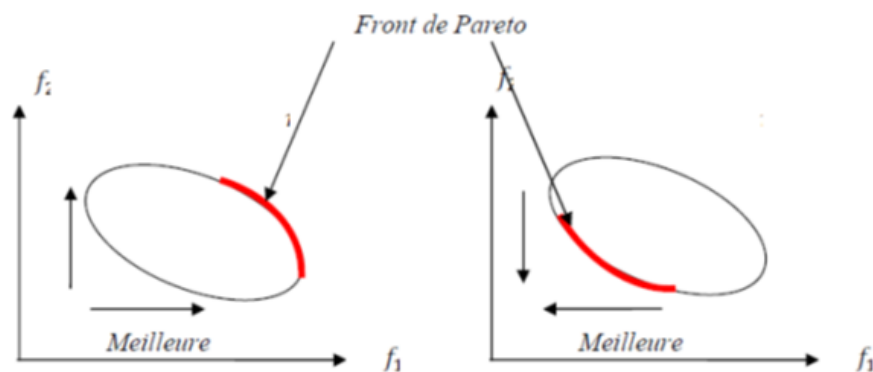


FIGURE 2.1 – Exemple de front de Pareto [1].

2.3 Méthodes de résolutions des MOPs

Afin de résoudre ce type de problème il existe plusieurs méthodes d'optimisation dans la littérature. Ces méthodes peuvent avoir une classification selon La fiabilité de la solution finale, le type de variable utilisée, type d'exploration de l'espace de recherche. Cette classification les classes en méthodes en méthodes déterministes et méthode méta-heuristique [177] (Figure 2.2).

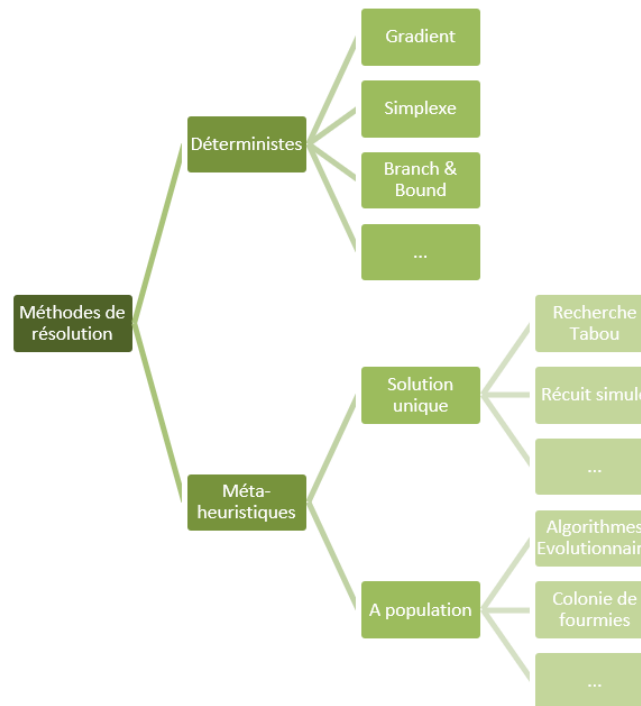


FIGURE 2.2 – Classification des méthodes d'optimisation.

2.3.1 Méthodes déterministe

Ce sont des méthodes qui se caractérisent par une exploration technique de l'espace de recherche. Elles s'exécutent dans un ordre prédéterminé (ex. : voie de recherche, point de départ (méthode d'origine)). Les algorithmes qui sont inclus dans la famille des méthodes déterministes : simplexe [178, 179], Quasi-newton [180], Gradient [179], Branch Bound [181].

Dans la pratique, il existe des MOPs très complexes pour lesquels la résolution par des méthodes déterministes peut être très longue ou peu réaliste, voire inopérantes. Il y a des exemples de problèmes où les fonctions objectifs ne sont pas fournies dans une

forme fermée, mais plutôt déterminées par un modèle de simulation. En outre, dans certains cas, le matériel à améliorer est directement utilisé pour évaluer une solution. Un autre trait commun aux problèmes de la vie réelle est la présence de bruit dans les fonctions objectives, ou des fonctions objectives dynamiques.

Par ailleurs, les méthodes déterministes présentent des limites dans la résolution des MOPs, que l'on résume dans les phrases suivantes :

- Ces algorithmes doivent être exécutés plusieurs fois afin de trouver plusieurs éléments de l'ensemble Pareto optimal.
- De nombreux MOPS nécessitent une connaissance approfondie du problème à résoudre.
- Certains algorithmes sont sensibles en fonction de la forme ou de la continuité du front Pareto.

De ce fait, le deuxième type des méthodes de résolution va nous permettre de surmonter ces limites qui sont les méthodes méta-heuristiques.

2.3.2 Méthodes méta-heuristiques

Les méthodes méta-heuristiques s'appuient sur des algorithmes qui guident et orientent le processus de recherche dans des espaces de solution qui sont souvent très vaste, en utilisant des régions riches en solutions optimales. L'abstraction et la généralisation de ces méthodes permettent une utilisation plus large pour des champs d'applications divers. Les méta-heuristiques sont en mesure de trouver des solutions pour ces applications, même si elles ne sont pas toujours optimales, mais plutôt proches de l'optimal et dans un délai raisonnable. Elles diffèrent ainsi des approches déterministes qui assurent certes la résolution d'un problème, mais au prix de temps de calcul excessifs [182].

Les méthodes stochastiques les plus importantes, qui s'appuient sur des notions de probabilité et des mécanismes aléatoires pour trouver la solution optimale, sont les méta-heuristiques. La capacité à converger vers l'optimum global est l'un des points forts de ces méthodes, sans grande influence sur le choix de la solution initiale ou du point de départ. Elles sont en général de zéro. Mais il faut parfois plusieurs générations

(itérations) de ces algorithmes pour trouver une solution dans le voisinage de l'optimum global [183].

L'objectif d'une heuristique est de trouver une solution appliquée sans garantir l'optimalité, contrairement aux méthodes déterministes qui garantissent des solutions précises. Il est préférable de faire appel aux heuristiques pour calculer une solution approchée d'un problème ou pour accélérer le processus de résolution exacte, car les algorithmes déterministes ont une complexité exponentielle pour les problèmes difficiles. En général, une heuristique est conçue pour un problème particulier, mais les approches peuvent comporter des principes plus généraux [184].

Les méta-heuristiques peuvent être classées [185] en fonction de leur utilisation d'une population de solutions et de leur utilisation d'une seule solution à la fois (figure). Les méthodes de recherche locale ou de trajectoire visent à améliorer une solution de manière itérative. La méthode Tabou et le Recuit Simulé sont des exemples courants de méthodes de trajectoire. Ces approches forment un parcours dans l'espace des solutions en cherchant à atteindre des solutions optimales. Par contre les EAs [186] sont l'exemple le plus célèbre des méthodes travaillant avec une population de solutions. Dans la partie suivante nous allons présenter les EAs, leurs principes de fonctionnement, ainsi que leur utilisation dans l'optimisation multi-objective.

2.4 Algorithmes évolutionnaires (Evolutionary Algorithms, EAs)

Dans le cas des problèmes d'optimisation ou il n'existe pas des méthodes exactes pour les résoudre, ou bien la solution optimale n'est pas préalablement définie, les chercheurs ont opté pour trouver des solutions approchées en utilisant des méthodes stochastiques d'optimisation globales comme les EAs. En s'inspirant librement de la biologie et de la théorie de Charles Darwin, les EAs dites bio-inspirées ont été créées. Les EAs servent à faire évoluer un ensemble de solutions à un problème donné pour arriver à une solution optimale. Ils sont appelés stochastiques car ils utilisent des processus aléatoires dans leur évolution. Ils sont généralement utilisés pour résoudre des problèmes d'optimisation ou bien pour trouver la meilleure solution.

Trois grandes familles des EAs ont été développées indépendamment dans l'histoire. Leur première apparition était depuis les années 1952 [187], où il y avait les premiers travaux pour utiliser les méthodes stochastiques dans l'optimisation, en 1965 Rechenberg a conçu le premier algorithme utilisant des stratégies évolutives [188] pour résoudre des problèmes d'optimisations continus. L'année suivante, Fogel, Owens et Walsh conçoivent la programmation génétique comme une méthode d'intelligence artificielle pour la conception d'automates à états finis. Puis, en 1975, Holland a formé les idées originales des algorithmes génétiques [189]. Et depuis les travaux ne cessent d'évoluer dans ce domaine comme l'apparition du calcul évolutionnaire (publié par les Massachusetts Institute of Technology) en 1993 qui est utilisée dans plusieurs domaines. Ces différentes approches ont beaucoup évoluées et se sont rapprochées, pour finir par être regroupées sous le terme générique des EAs (Figure 2.3).

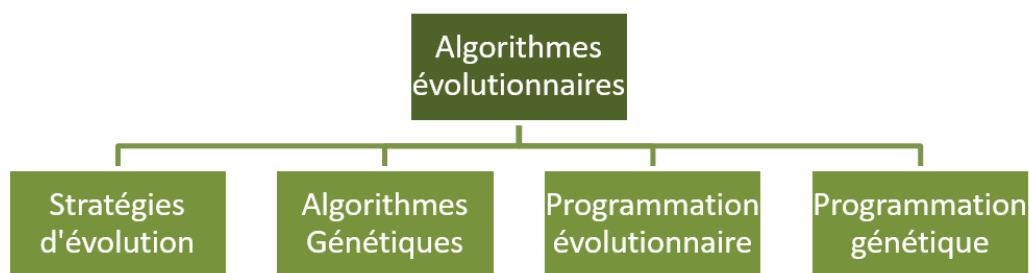


FIGURE 2.3 – Différentes familles des EAs.

Le principe de la théorie de l'évolution darwinienne peut être expliqué en présentant trois caractéristiques, la diversité, l'hérédité et la sélection naturelle [190]. Concernant la diversité, les traits (attributs) des individus appartenant à une population peuvent varier. Par conséquent, ces individus diffèrent les uns des autres dans une certaine mesure. Par exemple, par leur comportement ou leur apparence. De plus, pour l'hérédité, Certains traits sont systématiquement transmis des individus à leur descendance. Par conséquent, les descendants ressemblent davantage à leurs parents. Et dernièrement, la sélection naturelle se présente par le fait que les populations se battent généralement pour les ressources dans leur environnement donné. Les individus possédant les traits les mieux adaptés à l'environnement auront plus de chances de survivre et contribueront également à une plus grande descendance à la génération suivante. En d'autres

termes, l'évolution conserve ceux qui s'adaptent mieux à leur environnement et qui ont plus de chances de survivre, de se reproduire et de transmettre leurs caractéristiques à la génération suivante. En outre, un facteur important de l'évolution est le croisement ou la recombinaison, qui permet de créer une progéniture avec un mélange des caractéristiques de ses parents. Le croisement contribue à maintenir la diversité de la population et à rassembler les meilleurs traits au cours du temps. Cependant, les mutations peuvent jouer un rôle dans l'évolution en introduisant des variations aléatoires sur les caractéristiques d'un individu.

2.4.1 Composants de base d'un EA

Dans le but de réaliser l'optimisation d'un ensemble de solutions en appliquant le processus de sélection et de reproduction naturels, les EA impliquent plusieurs composants qui imitent ceux de l'évolution Darwinienne. Dans les sous sections suivantes, nous allons présenter les différents composants d'un EA.

2.4.1.1 Chromosome

La première étape de construction d'un EA est généralement la plus difficile car elle s'agit de choisir la bonne représentation pour le problème traité. Une représentation attribue des génotypes aux phénotypes correspondants. Cette phase permet de représenter une solution à un problème spécifique, le temps ou un même problème peut avoir différentes représentations. Les propriétés de la représentation utilisée, ainsi que d'autres paramètres comme l'adéquation et la variation des opérateurs génétiques, la taille de population [191], peuvent affecter la performance et l'efficacité de l'algorithme appliqué.

De nombreux types de représentations sont disponibles (Fig. 3). Les représentations linéaires sont présentées dans le codage binaire, le codage discret, le codage par permutation, Représentations quantiques, etc. Pour les codages non linéaires, les plus utilisés sont basés sur des structures de graphes, telles que des arbres.

2.4.1.2 Fonctions objectives

Les fonctions objectives se définies comme des métrique d'évaluation qui amènent la précision de l'algorithme génétique à la solution optimale [192]. Ces fonctions la quantifient la qualité et l'adéquation d'une solution individuelle au sein d'une population en lui affectant un poids appelé fitness. De ce fait, les fonctions objectives doivent être supportées et bien définies pour une meilleure convergence de l'EA. En évaluant la qualité des individus de la population, une comparaison aisée des individus peut être réalisée en utilisant une manière de les trier pour ne retenir que les meilleurs selon un certain nombre de critères (la sélection) [193].

Pour le succès d'EA, les fonctions de fitness doivent être bien conçues. Ainsi, afin d'en créer de bonnes fonctions, elles doivent être adaptées au problème spécifique que nous allons résoudre en capturant l'essence du problème et les principaux objectifs que nous voulons atteindre [194]. De plus, les fonctions conçues doivent être utiles pour faire la différence entre les bonnes et les mauvaises solutions en attribuant des scores élevés et faibles. De plus, ces fonctions doivent encourager l'exploration de l'espace de recherche et éviter une convergence prématurée afin d'aider l'EA à échapper aux optima locaux. Enfin, ces fonctions doivent être également efficaces sur le plan informatique pour éviter de ralentir l'algorithme.

2.4.1.3 Opérateurs génétiques

Les individus subiront un certain nombre d'opérations de variation pour imiter les modifications génétiques du gène, qui sont fondamentales pour trouver l'espace de solution. Les opérateurs de variation sont utilisés pour créer de nouveaux individus. L'idée sous-jacente est que la création d'un individu à partir de parents sélectionnés pour leurs qualités devrait pouvoir créer un meilleur enfant. Les opérateurs de variation prennent en entrée un certain nombre de parents et produisent un certain nombre d'enfants en sortie [193, 195, 196].

Un EA a deux classes distinctes d'opérateurs de sélection, la sélection pour la reproduction et la sélection pour la survie. Le choix de l'opérateur à imposer est discuté dans la discussion de contexte. Les deux opérateurs ont une base dans les théories de l'évolution de la biologie [197]. Une instance d'opérateurs de variation est l'opérateur

de croisement également appelé opérateur de recombinaison. Cet opérateur prend généralement multiples parents pour produire un ou plusieurs enfants après la recombinaison des gènes des parents. Le croisement doit tenir compte de la structure de l'individu. L'opérateur de mutation est peut-être l'opérateur unaire le plus basique de l'EA, qui prend généralement l'enfant nouvellement créé comme paramètre. Elle s'inspire de la mutation fondamentale observée en génétique biologique, en raison des différents effets des transcrits et des modifications spontanées des chromosomes observées dans les études évolutives primitives. La mutation vise à déplacer légèrement l'enfant dans l'espace de recherche pour éventuellement explorer des zones inaccessibles par un seul passage. Pour une discussion sur la façon dont les opérateurs génétiques peuvent intervenir avec EA, le lecteur dispose de plusieurs sources d'information dans la littérature, comme [198, 199] respectivement qui sont consacrées à résumer le domaine, discuter de larges types d'opérateurs.

2.4.1.4 Critères d'arrêt

La définition d'un critère d'arrêt dans un processus génétiques est considérée parmi les étapes les plus difficiles dans la création d'un EA. Car en éliminant le cas des problèmes artificiel, dans les problèmes réels nous ne pourrions jamais savoir si on a atteint une solution optimale car elle est déjà inconnue. De ce fait, généralement dans la pratique on opte pour fixer un nombre maximum de génération. Les autres critères d'arrêt se présentent dans l'arrêt de l'EA si la population cesse d'évoluer ou si on arrive à une population homogène. Un autre critère est si on atteint une solution optimale.

2.4.2 Principe de fonctionnement d'un EA

Pour faire fonctionner un EA on doit d'abord définir l'espace de solutions afin de créer la première population (groupe d'individu), le choix des représentations des individus (Phenotype) se fait dans la première étape de l'algorithme selon différentes structure : binaire, réel, par arbre. Ce qu'on appelle (Genotype).

La création de la fonction objective est basée sur les critères de la sélection des Genotypes qui forment la population. Le rôle d'une population est de représenter

les solutions possibles du problème qui doivent être variables pour assurer la présence d'une solution après les différentes opérations d'évolution. On se basant sur des critères de qualité, le choix des parents pour créer la prochaine progéniture se fait selon les résultats d'évaluation ou les meilleurs individus sont choisis pour être des parents de la nouvelle génération. Les opérateurs génétiques de modification (la mutation et la recombinaison) permettent la création des nouveaux individus à partir de ces parents choisis ou la mutation c'est l'introduction des valeurs pour les allèles qui n'étaient pas forcément présentes à un locus dans la population de départ [200] et la recombinaison c'est la fusion des informations des deux parents Génotypes dans un ou deux enfants Génotypes d'une façon stochastique. La taille des populations reste fixe tout au long la recherche donc un mécanisme de sélection des survivants doit être fait après avoir la nouvelle progéniture on se basant sur leur qualité d'évaluation pour créer la nouvelle population. Ce travail se reproduira jusqu'à la condition d'arrêt est franchie comme le montre la Figure 2.4.

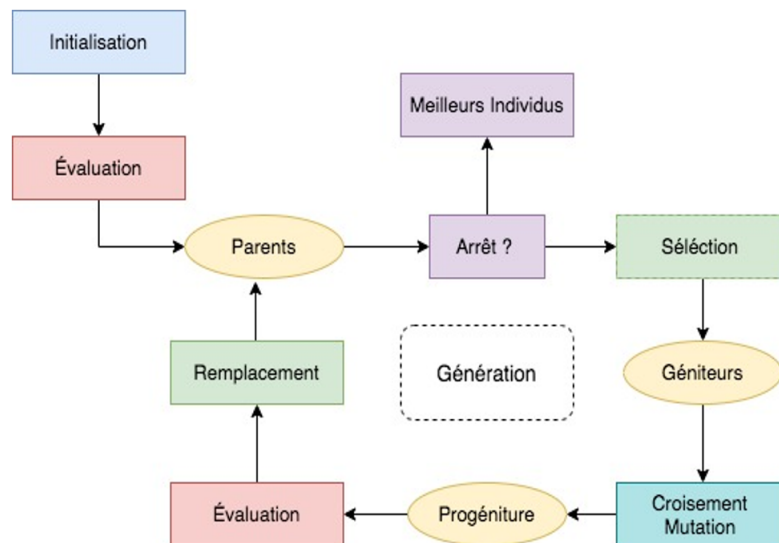


FIGURE 2.4 – Fonctionnement des EAs.

2.5 Algorithmes évolutionnaires multi-objectifs (Multi-objectif Evolutionary Algorithms, MOEAs)

La nature des EAs permet de traiter des MOPs, car Il est courant de constater qu'ils produisent des ensembles de solutions, ce qui permet de calculer une approximation de l'ensemble de la frontière de Pareto. De ce fait, de nombreuses études sur les EAs se sont concentrées sur les aspects multi-objectifs en utilisant des fronts de Pareto et une archive de solutions non-dominées [201]. La notion de Pareto dominance est employée par de nombreux EAs afin de classer les solutions et de définir des stratégies de sélection (de reproduction ou de survie). Il s'agit généralement d'algorithmes élitistes. Au fil des générations, ils conservent les meilleures solutions dans la population ou les conservent dans une archive.

Ces approches contribuent à la reproduction et peuvent ainsi influencer le déroulement de l'espace de recherche. Cependant, la croissance rapide du nombre de solutions non-dominées pour les MOPs peut entraîner une diminution du nombre de places disponibles (dans la population) pour de nouvelles solutions. Afin de remédier à cette situation, il est fréquent d'utiliser un opérateur de préservation de la diversité.

Les MOEAs sont particulièrement adaptés à la résolution des MOPs car ils fonctionnent simultanément avec un ensemble de solutions potentielles. Cette propriété permet de découvrir et d'exécuter avec une seule opération plusieurs solutions de l'ensemble optimal de Pareto (ou une bonne approximation). De plus, les MOEAs sont moins spectaculaires dans la forme ou dans la continuité du front de Pareto. Lorsque des méta-heuristiques voire les EAs sont appliquées pour résoudre des MOPs, le but devient d'obtenir une approximation de l'ensemble optimal de Pareto ayant deux propriétés : la convergence vers le front optimal de Pareto et la diversité uniforme [175]. La première propriété assure la génération de solutions de Pareto quasi-optimales, tandis que la seconde propriété indique une bonne distribution des solutions obtenues autour du front optimal de Pareto, de sorte qu'aucune information précieuse n'est perdue.

2.5.1 MOEAs basés sur la décomposition

Dans cette forme d'algorithme, le MOP est divisé en plusieurs sous-problèmes d'optimisation scalaires qui seront optimisés en même temps. En utilisant uniquement des informations provenant de ses plusieurs sous-problèmes voisins, chaque sous-problème est optimisé, ce qui réduit la complexité de calcul de MOEA/D à chaque génération par rapport à MOGLS et NSGA-II). Différentes méthodes sont disponibles pour transformer le problème de l'approximation de PF en différents problèmes d'optimisation scalaire. Tout comme la décomposition selon une méthode de somme pondérée, selon une méthode de Tchebyche [202].

2.5.2 MOEAs basés sur la dominance

Goldberg [203] a eu l'idée d'attribuer des aptitudes pour chaque critère afin que les résolutions en même temps même s'ils étaient contradictoires. Cette approche a permis de résoudre les MOP en utilisant des variantes de MOEA. Le premier algorithme qui a été proposé par Fonseca et Fleming [204] c'était un Algorithme Génétique multi-objectif (Multi-objectif Genetic Algorithm, MOGA). Dans ce Framework, Deux étapes essentielles ont été élaborées afin d'évaluer la conformité d'une solution. Tout d'abord, la forme physique d'une solution est déterminée en se basant sur le nombre d'autres solutions qui la surpassent. On utilise ce fitness pour évaluer la position des solutions. Ensuite, un mécanisme de partage de fitness permet de partager le fitness des solutions dans le même rang. Grâce à l'utilisation de ces deux étapes, MOGA est en mesure de conserver un ensemble de solutions non dominées lors d'une seule simulation.

Ensuite, Srinivas et Deb [205] ont proposé un autre MOEAs. Ils ont réfléchi à l'introduction d'un nouveau système de classement et de partage qui diffère de celui des MOGAs. On appelle l'algorithme proposé NSGA. Plutôt que d'évaluer le nombre d'individus qui gouvernent chaque solution, Srinivas et Deb ont suggéré un mécanisme de classification qui évalue le niveau de domination des solutions. Les solutions non dominées sont incluses dans le premier niveau. Par la suite, les solutions identifiées comme de premier ordre sont négligées. On identifie et définit la deuxième série de solutions non-dominantes dans la population comme le deuxième niveau (front). Cela se poursuit jusqu'à ce que les solutions stockées dans la population ne soient plus

disponibles pour le classement. La diversité est préservée grâce à une approche de partage de la condition physique.

Néanmoins, bien qu'ils soient efficaces dans la résolution des MOPs, le MOGA et le NSGA présentent malheureusement plusieurs contraintes. Tout d'abord, ils adoptent des méthodes non-élitistes. Ensuite, il est essentiel d'indiquer un paramètre de partage. Ensuite, l'algorithme NSGA présente une grande complexité de calcul. Pour améliorer l'efficacité de l'algorithme d'optimisations multi-objectives, certains chercheurs ont intégré un mécanisme d'élitisme, qui consiste en un archive de solutions externes, dans les algorithmes d'optimisation.

Comme mentionné dans [206], Zitzler et Thiele ont suggéré un MOEA élitiste nommé SPEA. Dans cet algorithme, la création d'un archive externe vise à conserver un ensemble de solutions non dominées (population externe) découvertes lors des processus évolutifs. Au cours de chaque génération, une nouvelle élite non dominée est identifiée dans la population actuelle, tandis que les solutions dominées dans la population externe seront rétrogradées. Dès que la population externe atteint un nombre maximal de solutions acceptées, les solutions surpeuplées, identifiées par un algorithme de regroupement, seront écartées. L'évaluation du fitness sera basée sur une valeur de fitness qui sera attribuée aux deux populations externes et actuelles. Le fitness d'une solution dans une population externe est lié au nombre de solutions présentes dans la population actuelle, où la solution dans la population externe est la plus dominante. D'un autre côté, la performance d'une solution dans la population actuelle est liée à la somme de performances des solutions dans la population externe qui prédomine la solution dans la population active.

Pour remédier aux problèmes des NSGA et SPEA, de nouvelles versions ont été établies comme l'algorithme NSGA-II [159] qui est l'un des plus réputés de cette catégorie. Quant à l'approche basée sur une archive de solutions non-dominées, elles ne contribuent pas forcément à la reproduction. Dans cette situation, la notion de dominance est souvent associée à d'autres critères pour évaluer les valeurs de fitness. Dans cette situation, la notion de dominance est souvent associée à d'autres critères pour évaluer les valeurs de fitness. Dans cette catégorie, Quant à lui, le SPEA II [207] est également une version améliorée du SPEA, avec une amélioration de l'affectation de fitness, l'archivage et un mécanisme de préservation de la diversité.

Dans la sous-section Algorithmes utilisés nous allons donner plus de détails sur les algorithmes NSGA II et le SPEA II et sur leurs principes de fonctionnement.

2.6 Problème d'optimisation de déploiement d'un WSN pour l'agriculture

Dans la nature, il existe de véritables problèmes d'optimisation qui reposent sur plusieurs critères souvent contradictoires et qui doivent être optimisés simultanément. Ainsi, pour optimiser les solutions à ces problèmes, nous devons trouver un vecteur de décision qui satisfait ces contraintes et optimise le vecteur objectif dont les éléments représentent des fonctions objectives ; c'est ce qu'on appelle l'optimisation multi-objectifs [208]. Les différentes fonctions objectives forment une description mathématique du critère de performance. Ces fonctions sont généralement en conflit. Le terme « optimiser » signifie trouver une solution où toutes les fonctions objectives renvoient des valeurs optimales en présence de certaines contraintes. Une solution peut être meilleure qu'une autre pour certains objectifs et moins bonne pour d'autres. Il n'existe donc généralement pas de solution unique offrant simultanément la solution optimale pour tous les besoins.

L'optimisation multi-objective traite plusieurs fonctions objectives en même temps. Pour cela, il utilise la notion de compromis optimal (une solution qui satisfait plusieurs fonctions à la fois). Les solutions sont séparées selon la notion de dominance au sens de Pareto [209].

Le déploiement du nœud capteur WSN dépend de plusieurs contraintes et doit satisfaire différents objectifs simultanément pour garantir les meilleures performances du WSN. Ainsi, comme dans le cas de l'AP, l'emplacement des nœuds dans les zones agricoles est lié au type d'application à surveiller, à la qualité et aux caractéristiques des nœuds capteurs, à la technologie de communication adoptée [210], etc.

Dans notre étude, nous avons souhaité trouver les meilleurs déploiements de nœuds capteurs pour une application de surveillance des sols en répondant au mieux aux différentes contraintes dans les plus brefs délais. Le modèle que nous proposons vise à satisfaire plusieurs objectifs à la fois comme le montre la figure 2.4 (celles qui sont

mentionnées en haut de la figure sont des contraintes classiques et celle du bas est la nouvelle contrainte qui a été proposé). Il choisit d'optimiser la couverture de l'ensemble de la zone agricole tout en maintenant la connectivité entre les différents nœuds capteurs et nœuds puits pour assurer le routage des données et la bonne qualité de service. De plus, le déploiement doit également minimiser la sur-couverture entre nœuds voisins et le coût du réseau exprimé en termes de nombre de nœuds déployés.

Une autre contrainte que nous trouvons importante lors du déploiement des nœuds dans un WSN dédié au PA, est la minimisation de la distance entre les nœuds capteurs. Parce que dans la surveillance des sols, si l'on veut modéliser les paramètres physiques à l'aide des données des nœuds installés, la distance séparant les nœuds peut affecter la qualité de ces modèles créés.

2.6.1 Algorithmes utilisés

Dans notre étude, nous avons utilisé deux types de variantes des MOEAs pour analyser leur efficacité dans la résolution du problème traité, chacune ayant ses propres caractéristiques et avantages. Cette approche nous a permis d'explorer les performances de chaque variante dans des scénarios spécifiques, en tenant compte des différents objectifs à optimiser.

2.6.1.1 Strength Pareto Evolutionary Algorithm II (SPEA II)

SPEA II propose par [175] est un MOEA qui est une extension de SPEA [211]. Les SPEA II font partie des fameux MOEAs. Ces dernières appliquent le principe du plus fort. Un ensemble de solutions évolue selon une ou plusieurs fonctions objectives, en appliquant des opérateurs génétiques de sélection, de croisement et de mutation. Gardant le même principe général de fonctionnement des EAs (Figure 3), ou le SPEA II intègre la notion de dominance de Pareto dans la phase de sélection.

À la différence entre le SPEA II par rapport au SPEA est qu'il adopte une méthode d'affectation de fitness qui intègre les données de densité. Avec SPEA, la taille de l'archive peut fluctuer au fil du temps, c'est-à-dire que chaque fois que le nombre d'individus non-dominés est inférieur à la taille prédéfinie, l'archive est remplie par

des individus dominés. La méthode de regroupement, utilisée lorsque le front non-dominé dépasse la limite d'archives, a été remplacée par une méthode alternative de troncature qui présente des caractéristiques similaires mais ne permet pas de perdre les points limites. Enfin, une autre distinction à SPEA réside dans le fait que seuls les membres de l'archive sont impliqués dans le processus de sélection.

Le SPEA II aide à trouver ou à se rapprocher de l'idéal de Pareto (une allocation de ressources sans alternative) fixé pour des problématiques d'améliorations multi-objectives. Les nouvelles données fournies par cet algorithme par rapport aux algorithmes génétique classique sont la création d'une archive dans laquelle l'algorithme stocke toutes les solutions non dominées à chaque itération et enfin la réduction du nombre de solution dans l'archive à l'aide d'une méthode de regroupement, ses étapes sont montrées sur la Figure 2.5.

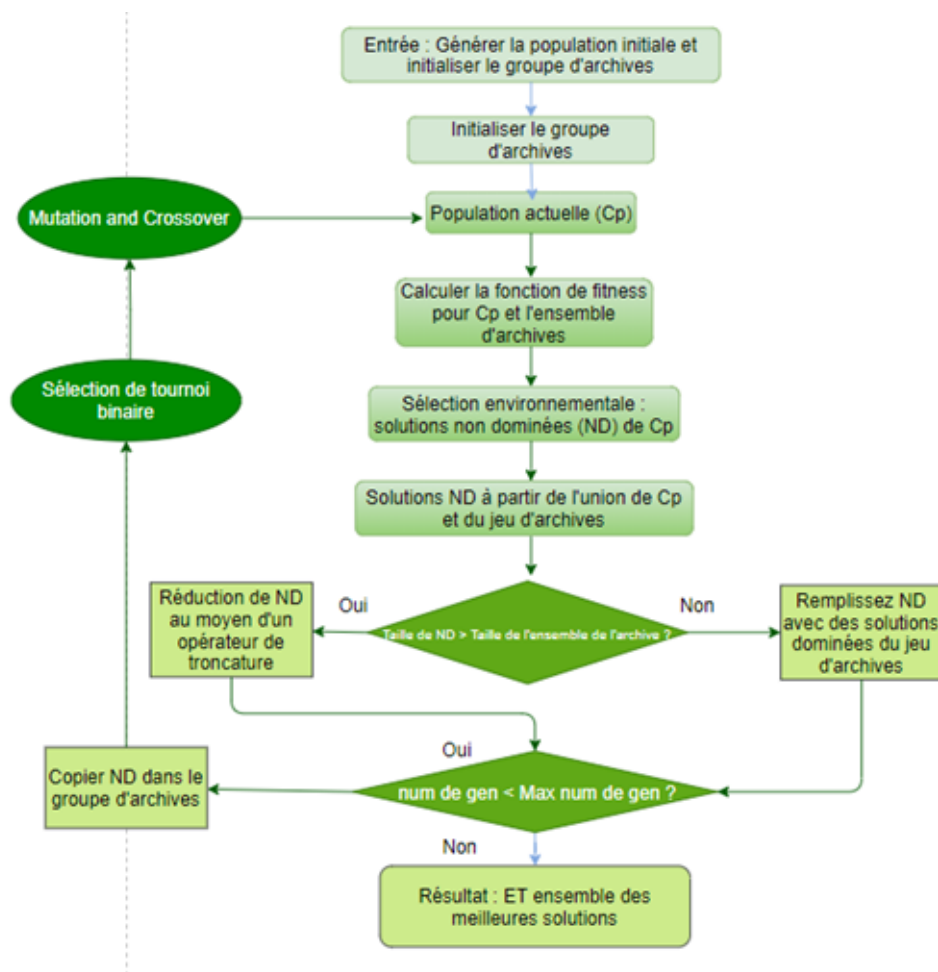


FIGURE 2.5 – Principe de fonctionnement de l'algorithme SPEA-II.

2.6.1.2 Non Sorting Genetic Algorithm II (NSGA II)

NSGA-II [205] a toujours été l'algorithme le plus utilisé dans le domaine de l'optimisation évolutionnaire multi-objective. Il tient son appellation de l'algorithme NSGA qui a été proposé auparavant par les mêmes auteurs [205]. L'algorithme NSGA reprend l'idée proposée par Goldberg sur l'utilisation du concept de classement par dominance dans les algorithmes génétiques [203]. Dans la plus part des aspects, NSGA-II est très différent de NSGA, cependant le nom a été gardé pour indiquer les origines de cette approche. Nous nous contenterons dans ce qui suit de présenter les différentes étapes de l'algorithme NSGA-II.

Dans NSGA-II la population des enfants Q_t est d'abord créée en utilisant la population des parents P_t . Les deux populations sont ensuite réunies pour former la population mixte R_t de taille $2N$. Cette population est triée selon le critère du rang de Pareto décrit ci-dessus pour former les fronts successifs : le premier front F_1 correspond à l'ensemble des solutions non-dominées de R_t . En considérant le reste des individus dans R_t après avoir enlevé ceux de F_1 , et après avoir réalisé un nouveau tri de dominance, nous obtenons le deuxième front F_2 constitué des individus non-dominés de l'ensemble $(R_t \setminus F_1)$. Cette procédure est répétée jusqu'à ce que tous les individus de R_t soient attribués à un front. Par la suite, la nouvelle population P_{t+1} est créée et est remplie au fur et à mesure avec les différents fronts successifs.

Comme la taille de la population R_t est $2N$, les fronts successifs ne peuvent pas intégrer en totalité la nouvelle population qui doit être de taille N . Ces fronts seront tout simplement éliminés. Cependant, la taille du dernier front considéré peut être supérieure aux nombres de cases vides à remplir dans la nouvelle population. Dans ce cas, le critère de surpeuplement décrit dans la section précédente sera utilisé pour choisir parmi les solutions du dernier front, celles qui vont intégrer la nouvelle population afin de favoriser les solutions dans les régions les moins peuplées du front comme le présente la figure 2.6.

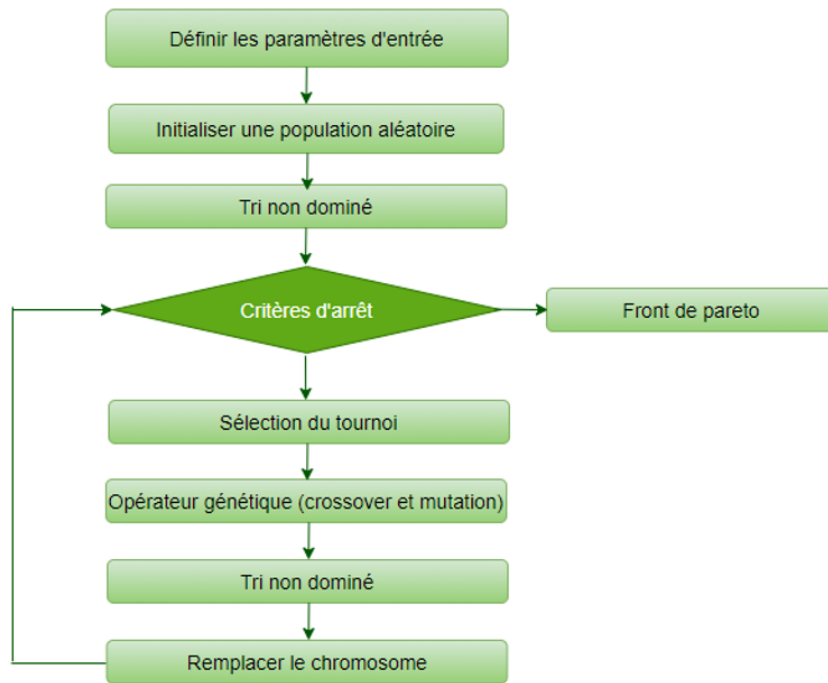


FIGURE 2.6 – Principe de fonctionnement de l'algorithme NSGA II.

2.6.2 Étude des travaux antérieurs

Plusieurs travaux de la littérature se sont intéressés à la problématique du déploiement de nœuds capteurs dans les WSN pour différents domaines d'utilisation tels que les bâtiments intelligents, les applications militaires, l'agriculture, etc. Dans notre étude, nous nous sommes intéressés aux travaux qui ont appliqué les GAs dans leurs modèles de déploiement. Grâce à la souplesse d'application des GAs et leur adaptation à de nombreux types de problèmes complexes dans la pratique [212]. Dans les problèmes d'optimisation, les GAs comme les EAs font évoluer un ensemble de solutions selon un nombre de générations déjà défini, où chaque solution est un candidat possible pour un optimum du problème d'optimisation. La représentation des solutions dans la famille GAs est sous forme de chaînes de valeurs ; ils sont appelés vecteurs dans le cas continu et chaînes de bits dans le cas des bits. Constamment, les GAs appliquent des opérateurs génétiques comme la sélection, le croisement et la mutation afin de garantir la diversité dans l'espace de recherche.

Dans le cas des problématiques de déploiement des WSN, où les contraintes de couverture, de connectivité et de durée de vie du réseau jouent un rôle primordial

dans la performance et le fonctionnement des WSNs [213], les GAs ont été largement utilisés, du fait de leur adaptabilité et de leur support d'optimisation multi-objectif [214]. Dans la littérature, nous avons constaté que les GAs à objectif unique (SOGAs), les GAs bi-objectifs (BOGAs) et les GAs multi-objectifs (MOGAs) ont été appliqués pour optimiser le positionnement des nœuds des WSNs afin de répondre à différents objectifs, comme indiqué dans le tableau 2.1. Cependant, La Figure 2.7 illustre une comparaison entre les 16 travaux présentés dans le tableau 1 selon le nombre d'objectifs pris en compte pour le déploiement des WSN. Nous avons constaté que 43,75 % étaient des SOGAs, 25 % étaient des BOGAs et 31,25 % étaient des MOGAs.

La différence entre ces trois types réside dans le nombre de contraintes à prendre en compte lors du déploiement des nœuds. Comme plusieurs paramètres doivent être optimisés simultanément et peuvent potentiellement entrer en conflit les uns avec les autres. Par exemple, dans chaque déploiement WSN, les capteurs doivent assurer la couverture la plus élevée de la zone intéressée avec le moindre coût (nombre de capteur). De plus, la connectivité et la couverture du réseau peuvent être maximisées en même temps, mais cette maximisation risque d'influencer la durée de vie du réseau. Il est donc remarquable que plus on diminue le nombre de contraintes plus les GAs deviennent efficaces en terme de temps de calcul, mais la qualité des solutions trouvées dans le cas des SOGAs [215], [216], [159], [217], [218], [219], [220] et BOGAs [221], [222], [223], [224] ne peut pas satisfaire toutes les contraintes d'exigence qui affectent les emplacements des capteurs comme les solutions MOGAs [225], [226], [227], [228], [229].

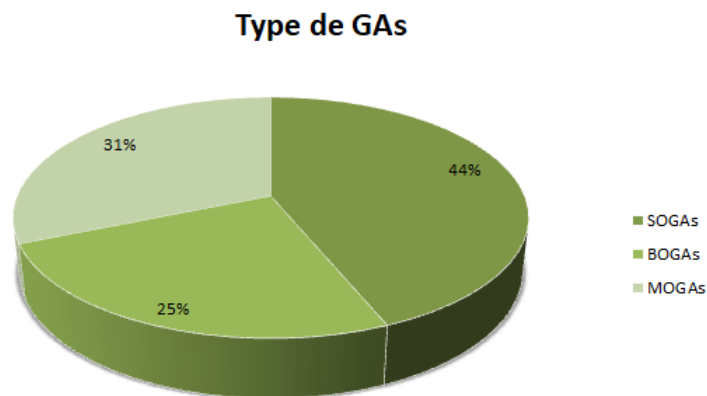


FIGURE 2.7 – Comparaison des types de GA pour le déploiement WSN.

TABLE 2.1 – Applications des GAs pour l'optimisation du déploiement des WSNs.

Travaux (année)	GA utilisé	Objectifs	Type de GA
Hoffmann, Medina, et Wolisz. (2011) [215]	GA hybride	Minimisation de la latence	SOGA
Tripathi et al. (2013) [221]	GA et programmation génétique	Couverture et maximisation de la durée de vie du réseau	BOGA
Yoon et Kim. (2013) [216]	GA	Maximisation de la couverture avec un rayon de détection variable de différents capteurs.	SOGA
Rebai et al. (2015) [222]	GA	Couverture et maximisation de la connectivité	BOGA
Gupta, Kuila et Jana. (2016) [159]	GA	Maximisation de la connectivité	SOGA
Gupta, Kuila et Jana. (2017) [225]	MOGA	Maximisation de la couverture et de la connectivité et nombre minimum de nœuds de capteurs	MOGA
Dai et Wang. (2017) [223]	GA amélioré	Maximisation des informations connectées et de la couverture	BOGA
Benatia et al. (2017) [226]	MOGA	Maximisation de la couverture et de la connectivité et minimisation du nombre de nœuds et de la surcouverture	MOGA
Karatas. (2018) [217]	Schéma basé sur GA	Couverture hybride de WSN hétérogènes	SOGA
Liang et Lin. (2018) [218]	GA	Maximisation de la stratégie de couverture dans MWSN	SOGA
Panhwar et al. (2018) [219]	GA	Optimisation de l'énergie basée sur la distance	SOGA
Perez. (2018) [224]	NSGA II	Nombre total d'appareils utilisés dans le placement et énergie totale dissipée par le placement	BOGA
Hanh et al. (2019) [220]	MIGA et VFA	Maximisation de la couverture de zone	SOGA
Harizan et Kuila. (2019) [227]	GA amélioré	Maximisation de la connectivité et de la couverture et minimisation de l'énergie	MOGA
ZainEldin et al. (2020) [228]	Déploiement dynamique améliorée basée sur GA (IDDT-GA)	Maximisation de la couverture avec le plus petit nombre de nœuds et minimisation de la zone de chevauchement	MOGA
Pal et al. (2021) [229]	NSGA II	Force du signal reçu, couverture et surcouverture	MOGA

Les travaux présentés visent tous à trouver le déploiement optimal des nœuds capteurs en fonction des contraintes définies. Cependant, il existe une différence entre eux dans les objectifs recherchés. La Figure 2.8 présente le pourcentage de contraintes de déploiement des WSNs extraites des travaux précédents. Selon cette étude, nous avons constaté que la maximisation de la couverture et de la connectivité étaient les objectifs les plus recherchés avec le pourcentage de 75% et 50% respectivement, suivis de la maximisation de l'objectif d'efficacité énergétique avec 25%, que la minimisation du capteur nombre de nœuds avec 18,75% et enfin la minimisation de la zone de sur-couverture avec 12,5%.

Les travaux cités ont prouvé l'avantage et la possibilité d'appliquer les GAs pour l'optimisation du déploiement des nœuds capteurs dans les WSNs selon des fonctions de fitness uniques ou multiples dans différents domaines d'utilisation, comme pour l'agriculture. Le choix du type de GA à utiliser dépend du nombre de contraintes prises en compte lors du déploiement du WSN. Par ailleurs, le temps de calcul et les qualités de la solution sont également liés au nombre de contraintes. Dans notre étude, puisqu'elle n'a pas été testée auparavant, nous avons cherché à comparer le fonctionnement de deux variantes de GAs multi-objectifs selon 5 contraintes en même temps pour une application agricole.

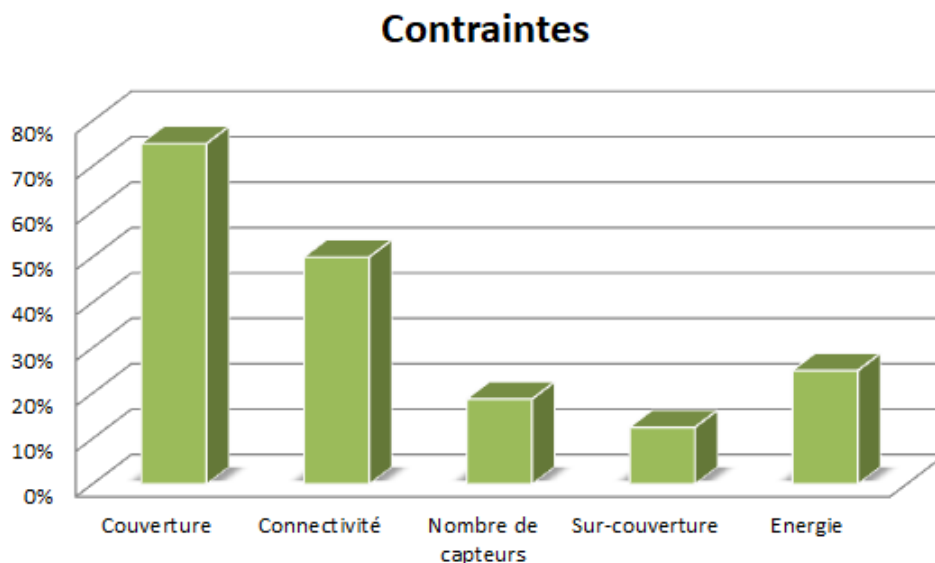


FIGURE 2.8 – Pourcentage d'utilisation des différentes contraintes dans les 16 ouvrages étudiés.

2.7 Adaptation des algorithmes MOEAs au problème de déploiement

Afin de résoudre ce problème multi-objectif, nous avons appliqué deux variantes des MOEAs, le SPEA II et le NSGA II. Dans les sous-sections suivantes, nous présenterons et expliquerons chaque étape de notre modèle proposé pour le déploiement des nœuds d'un WSN pour une application de surveillance des sols.

2.7.1 Modélisation de l'espace de déploiement

Considérons une parcelle agricole de surface plane (dans notre exemple nous suggérons une forme rectangulaire à surveiller à l'aide d'un WSN. Les nœuds capteurs et les nœuds puits sont déployés de manière déterministe de sorte que chaque point du tracé est surveillé par au moins un nœud capteur, et il existe un chemin, composé d'un nombre limité de sauts de chaque nœud capteur au puits. L'espace de déploiement du site est discrétisé et modélisé comme une grille rectangulaire à deux dimensions longueur (L) et largeur (l) (Figure 2.9), où chaque cellule ($C_{x,y}$) a une superficie de $1m^2$. La difficulté d'exploiter toutes les positions possibles dans une zone bien déterminée nous oblige à déployer les nœuds capteurs \ puits aux centres des cellules qui ont l'indice I . Ces centres seront utilisés par la suite pour calculer les différents critères de déploiement. Par exemple, pour calculer la distance entre deux nœuds capteurs, on calcule la distance entre les deux centres de cellule où sont placés ces deux nœuds.

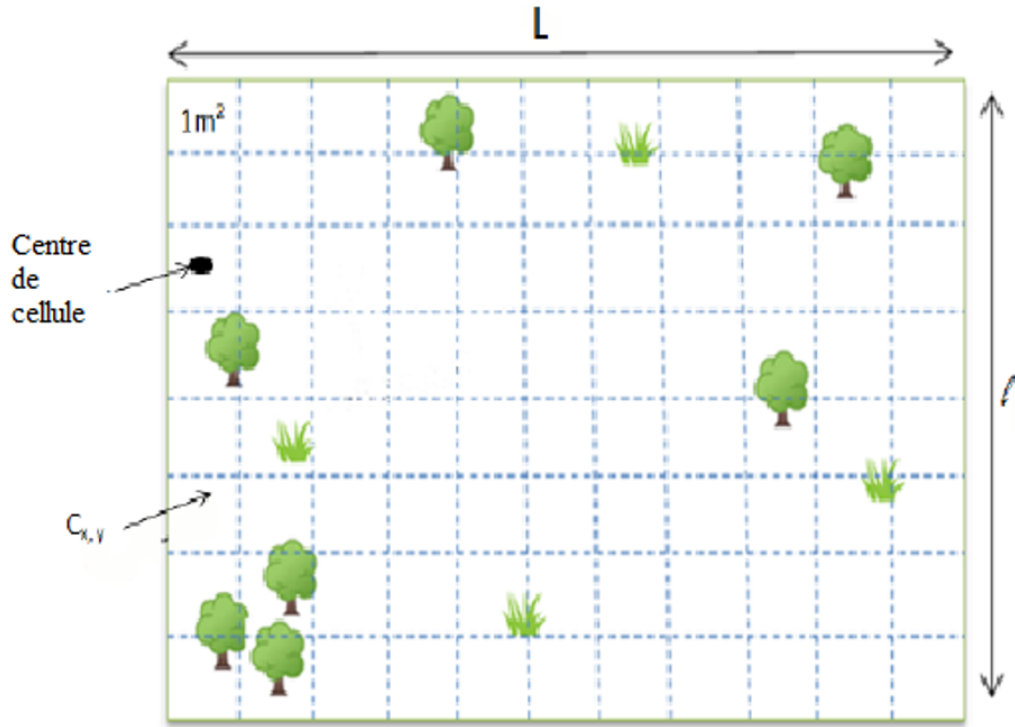


FIGURE 2.9 – Grille de déploiement des nœuds.

2.7.2 Initialisation de la population

Les algorithmes MOEAs s'adaptent facilement à la représentation binaire. Par conséquent, nous avons choisi d'utiliser le codage binaire dans la formulation des chromosomes. Nous avons utilisé un vecteur de taille fixe, résultant de la translation de la matrice qui représente la grille de déploiement pour représenter les variables de décision qui sont elles aussi binaires à leur tour (Figure 2.10). Chaque position notifie la présence ou l'absence d'un nœud dans une cellule $(C_{x,y})$, ce qui facilite l'application d'opérateurs de mutation et de croisement par la suite. La valeur 1 signifie la présence d'un nœud dans cette position et le 0 signifiant son absence.

La population initiale dans notre cas est un ensemble de solutions générées aléatoirement pour garantir la diversité de la population initiale afin d'éviter de tomber dans le piège des optimums locaux. La création d'un individu est la suivante : un capteur est ajouté aléatoirement à la grille jusqu'à ce que la grille soit entièrement recouverte comme mentionné dans le Tableau 2.

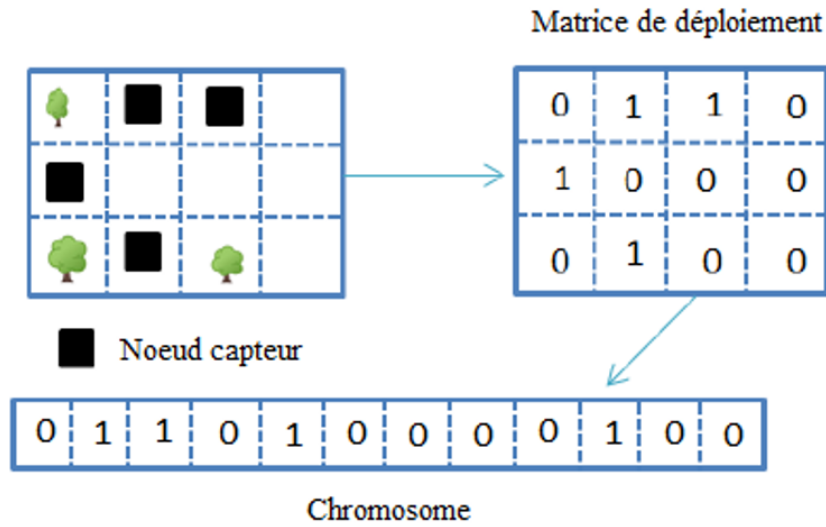


FIGURE 2.10 – Formulation et codage des chromosomes.

Individuels	Chromosomes
1	10010110
2	01000101
3	10011110
4	10111111
5	01010001

FIGURE 2.11 – Exemple d'un individu généré aléatoirement de la population initiale (rendement de la dimension 8 m²).

2.7.3 Modélisation des fonctions objectives

Notre problème de déploiement est un problème multicritère. Le réseau de capteurs installé doit simultanément maximiser la couverture et la connectivité, minimiser le nombre de capteurs utilisés, les zones de sur-couverture et la distance séparant les nœuds. Sachant que les nœuds puits ont de meilleures caractéristiques énergétiques que le reste des nœuds capteurs, leur évaluation se fait selon 5 fonctions objectives. Par conséquent, un modèle mathématique est conçu pour chaque critère de déploiement. Ces modèles représentent les données d'entrée pour la phase d'optimisation du placement des nœuds. La bonne prise en compte de tous les paramètres liés au déploiement du WSN permet d'obtenir des résultats satisfaisants les besoins des utilisateurs.

2.7.3.1 Nombre de nœuds de capteur

Le nombre de capteurs déployés affecte le coût d'installation du réseau. Sachant que le coût de mise en place du WSN est le principal paramètre à prendre en compte lors du choix du déploiement du réseau. Le déploiement des nœuds doit donc maximiser les contraintes de couverture et de connectivité tout en minimisant au maximum le nombre de nœuds déployés qui s'exprime à l'aide de l'équation suivante (Équation 2.1).

$$N_T = N_S + N_{SK} \quad (2.1)$$

N_T représente le nombre total de nœuds,

N_s représente le nombre de nœuds de capteurs,

N_{sk} représente le nombre de nœuds récepteurs.

Selon la modélisation de l'espace de déploiement, on peut définir la matrice de déploiement M comme suit :

$$M(x, y) = \begin{cases} 1 & \text{si un nœud capteur est déployé à la position (x, y)} \\ 0 & \text{autre} \end{cases} \quad (2.2)$$

Ensuite, la somme des numéros de nœuds peut être calculée comme dans l'équation 2.3, et le taux du nombre de nœuds de capteurs peut être calculé comme dans l'équation 2.4.

$$N_s = \sum_x \sum_y M(x, y) \quad (2.3)$$

$$N_{sr} = \frac{N_s}{L \times l} \times 100 \quad (2.4)$$

2.7.3.2 Distance de séparation entre les capteurs

En agriculture, on ne peut pas placer un capteur à chaque point de la parcelle. Par conséquent, pour connaître la valeur des paramètres physiques en tout point de la zone d'intérêt, nous devons réaliser un modèle continu à partir des données discrètes du sol. Dans ce cas, il faut utiliser un algorithme de Machine Learning (ML) du type supervisé qui doit appliquer la régression pour pouvoir prédire les valeurs des paramètres physiques.

Plusieurs algorithmes ML peuvent être appliqués pour la régression, comme les données prévoyant des régressions linéaires multiples qui sont les plus utilisées selon [230] et présentent de meilleurs résultats en prédiction par rapport aux autres algorithmes [231]. Cependant, la régression de processus gaussien n'est qu'une généralisation naturelle d'une régression linéaire multiple qui nous permet de proposer des limites de décision non linéaires [44] pour définir la distribution des fonctions non linéaires et linéaires et a la capacité inhérente de modéliser l'incertitude du modèle.

Selon Zhang et al [232], plus la distance qui sépare les nœuds capteurs est grande, plus la qualité des estimations est mauvaise. De plus, la présence de trous de couverture dans le réseau est directement liée à son erreur quadratique moyenne. De ce fait, la distance entre les nœuds capteurs doit être la plus minimale possible pour améliorer la qualité des modèles de prévision par un processus gaussien. Nous devons calculer la distance de séparation entre tous les capteurs déployés dans le WSN. Pour cela, nous avons utilisé la formule de distance euclidienne (équation 2.5) . Cette contrainte nous permet de plus de minimiser la consommation de l'énergie, avoir moins de zones de faible couverture et améliorer les communications.

$$Dist = \sum_{N_s} \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2} \quad (2.5)$$

(x_k, y_k) , les coordonnées du capteur K,

(x_i, y_i) , les coordonnées des autres capteurs.

2.7.3.3 Couverture et sur-couverture

Considérée comme l'une des métriques de Qualité de Service (QoS) dans notre modèle de déploiement, nous nous sommes intéressés au type de couverture de zone. Notre choix est basé sur les besoins coûteux pour détecter et quantifier les insectes souterrains à l'aide de capteurs acoustiques qui utilisent les sons ou les vibrations des insectes lorsqu'ils communiquent, et leurs bruits lorsqu'ils se nourrissent ou se déplacent. Par exemple, le système à ultrasons haute fréquence testé pourrait détecter à une distance allant jusqu'à 3 m (Portée de détection (Rd)). Seule la distance séparant le capteur de l'événement est prise en compte dans le calcul de la probabilité de détection. Ce dernier est supposé égal à 1 si la distance séparant le capteur de l'événement est

inférieure à la plage de détection R_d (portée de détection), et à 0 si l'événement se produit à une distance supérieure à R_d . Avec cette définition, nous pourrions définir la matrice de couverture de détection telle que présentée dans l'équation 2.6.

$$Det(x, y) = \begin{cases} 1 & \text{if } (C_{x,y}) \text{ est couvert par au moins un capteur} \\ 0 & \text{autre} \end{cases} \quad (2.6)$$

La fonction qui calcule le taux de couverture de détection peut être calculée en appliquant l'équation 2.7.

$$C_{vr} = \frac{\sum_x \sum_y Det(x, y)}{L \times l} \times 100 \quad (2.7)$$

En continuant la même procédure, nous pourrions définir la matrice de sur-couverture, où la cellule est couverte par plus d'un seul capteur en utilisant l'équation 2.8.

$$O_v - Det(x, y) = \begin{cases} 1 & \text{if } (C_{x,y}) \text{ est couvert par au moins 2 capteurs} \\ 0 & \text{autre} \end{cases} \quad (2.8)$$

Et donc l'équation 2.9, nous a permis de calculer le taux de sur-couverture :

$$O_v - C_{vr} = \frac{\sum_x \sum_y O - Det(x, y)}{L \times l} \times 100 \quad (2.9)$$

2.7.3.4 Taux de communication des nœuds

Dans notre modèle, nous nous concentrerons sur une topologie point à point du réseau basé sur la technologie IEEE 802.15.4 Zigbee, où tous les nœuds peuvent communiquer entre eux, parce que la technologie Zigbee est l'outil de communication le plus adopté en agriculture, grâce à ses propriétés telles que la portée de communication qui varie entre 10 et 100 m, et sa faible consommation d'énergie [102, 103].

En conséquence, dans notre étude, nous supposons que tous les nœuds ont la même portée de communication (R_c) avec de meilleures compétences énergétiques que les nœuds puits. La différence entre eux peut apparaître dans le mode de fonctionnement. Les nœuds puits sont censés agréger les données et les nœuds capteurs visent à détecter les phénomènes et à envoyer un rapport au nœud puits le plus proche. Cependant, pour modéliser la connectivité de la zone, nous avons dû utiliser le modèle de propagation en espace libre de Friis [230] pour calculer la perte de trajet (PL) du signal en appliquant l'équation 2.10. Ce modèle présente les avantages de rapidité et d'adaptation aux

espaces libres.

$$P_L(db) = -10\log_{10}[\lambda^2/(4\pi d)^2] = 20\log_{10}[4\pi d/\lambda] \quad (2.10)$$

P_L représente la perte de trajet du signal émis,

λ représente la longueur d'onde de la porteuse en mètres, d représente la distance de séparation entre l'émetteur et le récepteur en mètres.

Pour calculer la connectivité, nous avons pris en compte la perte de chemin du signal envoyé. Par conséquent, nous avons utilisé l'équation 2.10 pour calculer le (PL) du signal qui nous a permis de calculer le RSSI comme mentionné dans l'équation 2.11 ci-dessous.

$$RSSI(x, y) = P_T - P_L \quad (2.11)$$

P_T énergie de transmission.

On déduit de cette équation la matrice de connectivité $Conc(x,y)$ où chacun de ses éléments peut prendre 1 si le RSSI calculé dans la cellule $C(x,y)$ est supérieur à P_m (le RSSI minimum nécessaire pour avoir une bonne communication), et 0 sinon comme illustré dans l'équation 2.12.

$$C_n(x, y) = \begin{cases} 1 & \text{si la position RSSI}(x, y) \text{ est supérieure à } P_m \\ 0 & \text{autre} \end{cases} \quad (2.12)$$

Ensuite, nous pouvons modéliser la fonction du taux de connectivité comme suit (Équation 13) :

$$C_{on} = \frac{\sum_x \sum_y C_n(x, y)}{L \times l} \times 100 \quad (2.13)$$

Par conséquent, la fonction de fitness englobe nos cinq fonctions objectives (N_{sr} , $Dist$, C_{vr} , $O_v - C_{vr}$, C_{on}). La fonction d'évaluation aura un individu comme paramètre et produira cinq valeurs, qui représentent la valeur de fitness pour chaque contrainte.

2.7.4 Opérateurs génétiques

2.7.4.1 Sélection des parents

La phase de sélection définit le nombre de participations faites pour un individu à la reproduction dans une génération. Les individus avec de meilleures valeurs de fitness sont sélectionnés plus fréquemment que les autres. Les individus sélectionnés

sont appelés parents et sont choisis pour participer à la phase suivante appelée reproduction. Dans notre cas, la sélection des individus qui participent au croisement et à la mutation s'effectue selon la méthode de sélection par tournoi. Deux parents sont tirés au sort parmi la population et participent à un tournoi. Le parent ayant la meilleure valeur de fitness est sélectionné. Si les deux parents sélectionnés ont la même valeur de fitness, une autre métrique est ajoutée : l'individu avec la valeur minimale connue sous le nom de "distance d'encombrement" est sélectionné (Figure 2.12).

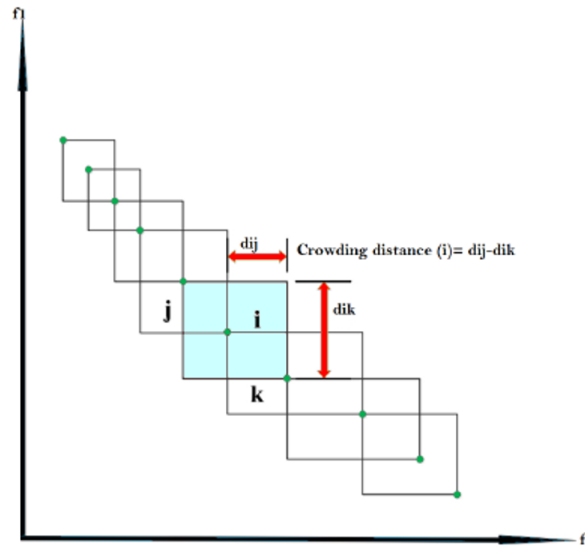


FIGURE 2.12 – Distance d'encombrement.

2.7.4.2 Opérateur de croisement

Après avoir sélectionné les parents qui participeront à la production des nouveaux individus, un opérateur de croisement doit être appliqué à ces parents avec une certaine probabilité de croisement. Le taux de croisement (probabilité de croisement) définit la proportion de parents dans la population qui sera utilisée par un opérateur de croisement. Au début de l'algorithme, le croisement joue un rôle d'exploration dans l'espace de recherche, car les nouvelles solutions générées par le croisement se retrouvent dans des espaces non explorés auparavant. Dans notre modèle, nous proposons d'utiliser l'intersection en un point comme le montre la Figure 2.13.

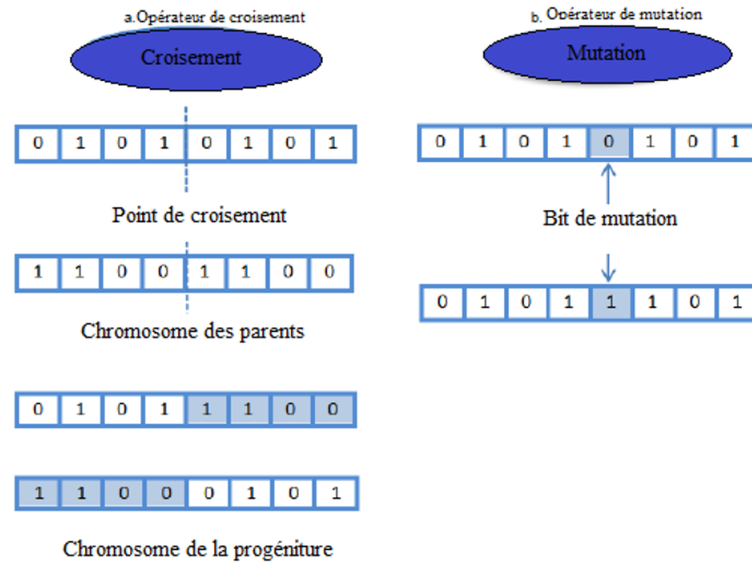


FIGURE 2.13 – Opérateur de croisement et de mutation.

2.7.4.3 Opérateur de mutation

Une fois l'opérateur de croisement est appliqué aux parents, c'est au tour de l'opérateur de mutation d'introduire une perturbation dans la solution avec une faible probabilité (PM) appelée taux de mutation. L'application de cet opérateur garantit la diversité de la population et permet également de mieux explorer l'espace de recherche, ce qui évite de converger vers des optima-prémisses. Puisque nous avons proposé d'utiliser un codage binaire des individus, nous proposons d'utiliser une méthode binaire de mutation (Figure 13). La valeur d'un gène dans le chromosome (l'individu), choisi au hasard, est inversée (0 si c'était 1, et 1 sinon). Ce type de mutation reste à notre connaissance le type le plus adapté pour les variables binaires.

Cependant, pour les probabilités de mutation (PM) et de croisement (PC), nous les avons fixées respectivement à 0,1 et 0,9, car plus les deux probabilités de croisement et de mutation sont importantes, plus les résultats sont efficaces malgré que le temps d'exécution soit proportionnel aux probabilités.

2.7.5 Critères d'arrêt

Pour obtenir de bons résultats, le critère d'arrêt dans notre modèle est de fixer le nombre de générations à atteindre au cours de l'algorithme. Car dans notre cas, nous

n'avons pas de modèle de déploiement déjà défini et nous n'avons pas des préférences à une contraintes par rapport au autres contraintes. Donc pour les critères d'arrêt de notre modèle nous avons choisi de fixer le nombre de générations qui est lié à la taille de la parcelle agricole à surveiller. Plus la taille de la parcelle est grande, plus le nombre de génération doit être élevé, pour mieux explorer l'espace de recherche.

De plus, les résultats de notre modèle sont représentés par un front de Pareto comme sur la Figure 2.14. Les topologies de réseau proposées varient entre un faible coût de déploiement, une sur-couverture minimale (avec une connectivité et une couverture faibles) jusqu'à un nombre élevé de nœuds déployés (coût élevé), qui assurent une couverture totale de l'espace avec une excellente connectivité (100%), mais en même temps, un sur-dénombrement élevé. Toutes les solutions proposées par les MOEAs sont exploitables. Le choix est laissé à l'utilisateur et dépend de ses besoins.

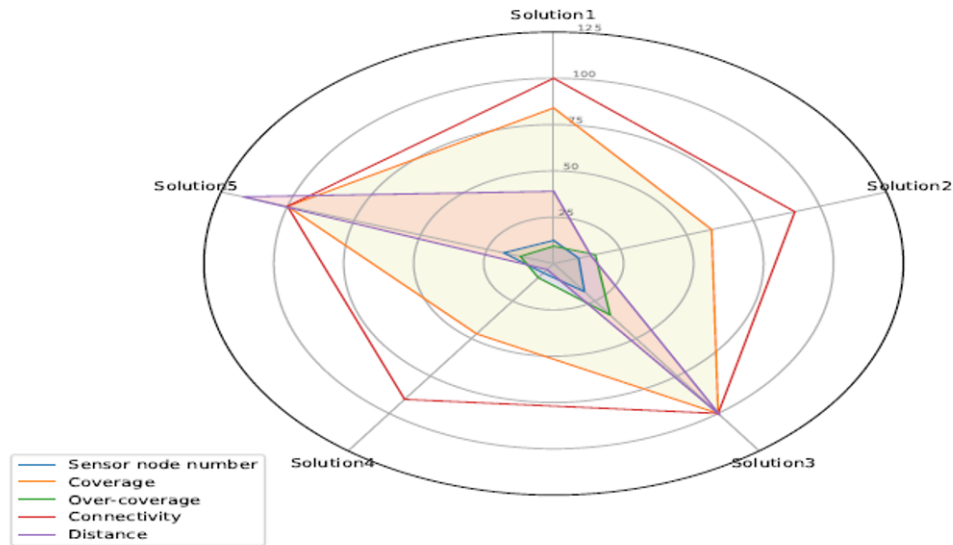


FIGURE 2.14 – Front de Pareto d'un ensemble de solutions pour un exemple de taille de parcelle de 32m².

2.8 Comparaison entre SPEA II et NSGA II

Pour établir une comparaison efficace entre deux types d'algorithmes d'optimisation multi-objectifs, à savoir le NSGA-II et le SPEA-II, il est essentiel de définir des

paramètres appropriés. Ces paramètres faciliteront l'évaluation et permettront de mesurer objectivement les performances des deux algorithmes. Voici comment nous avons procédé.

2.8.1 Définition des hyper-paramètres MOEAs

Afin d'établir une comparaison entre les deux variantes de MOEAs qui sont le NSGA II et le SPEA II, nous avons fait des tests expérimentaux pour pouvoir définir les bons hyper-paramètres des MOEAs comme le nombre de générations et la taille des populations. Or, ces deux paramètres dépendent en premier lieu de la taille de la parcelle agricole, plus elle est grande plus ces deux paramètres sont élevés pour pouvoir exploiter l'espace de recherche, donc nos chromosomes auront une longueur de 100 bits. De plus, pour définir les hyper-paramètres, nous avons fait varier la taille de la population de 10 à 80 avec un pas de 10 et des générations de 1 à 80. De ce fait, nous avons trouvé que le temps d'exécution est proportionnel à la taille de la population comme le montre la Figure 2.15 et que notre modèle commence à atteindre un optimum global avec 60 individus dans la population à la génération 70, et l'atteint avec 70 et 80 individus comme le montre également la Figure 2.15. Par conséquent, nous avons décidé de fixer la taille de la population à 70 individus afin de garantir la convergence de notre modèle dans un temps d'exécution minimum, et le nombre de génération à 100 afin de donner plus d'opportunité à notre modèle d'explorer l'espace de recherche.

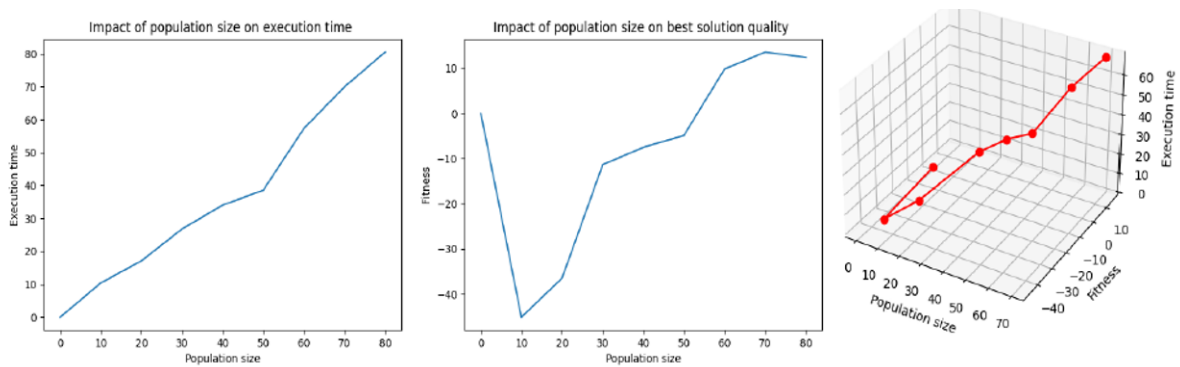


FIGURE 2.15 – Impact de la taille de la population sur la qualité de la meilleure solution et le temps d'exécution.

2.8.2 Résultats et discussion

Après avoir défini les différents hyper-paramètres des MOEAs, nous avons évalué le fonctionnement de NSGA II et SPEA II pour une parcelle de 100 m^2 , et nous avons évalué leurs performances en comparant la valeur de fitness et l'efficacité temporelle des deux variantes des MOEAs. Les deux algorithmes ont été implémentés à l'aide de Python, et ont été exécutés sur un PC avec un processeur Intel Core i5-4300U, 2,50 GHz et 4,00 Go de RAM afin d'avoir une comparaison équitable entre eux.

La comparaison entre les deux variantes : SPEA II et NSGA II se base sur plusieurs objectifs : le nombre de capteurs, la couverture, la sur-couverture, la connectivité et la distance totale. La Figure 2.16 nous montre l'évolution des 5 fonctions objectives par rapport aux différentes générations pour les deux algorithmes. De plus, la Figure 2.17 présente une comparaison entre les solutions non dominées NSGA II et SPEA II selon leurs fonctions objectives et le tableau 2.2 présente Comparaison entre SPEA II et NSGA II en fonction de la qualité des solutions et du temps d'exécution (en secondes) [233].

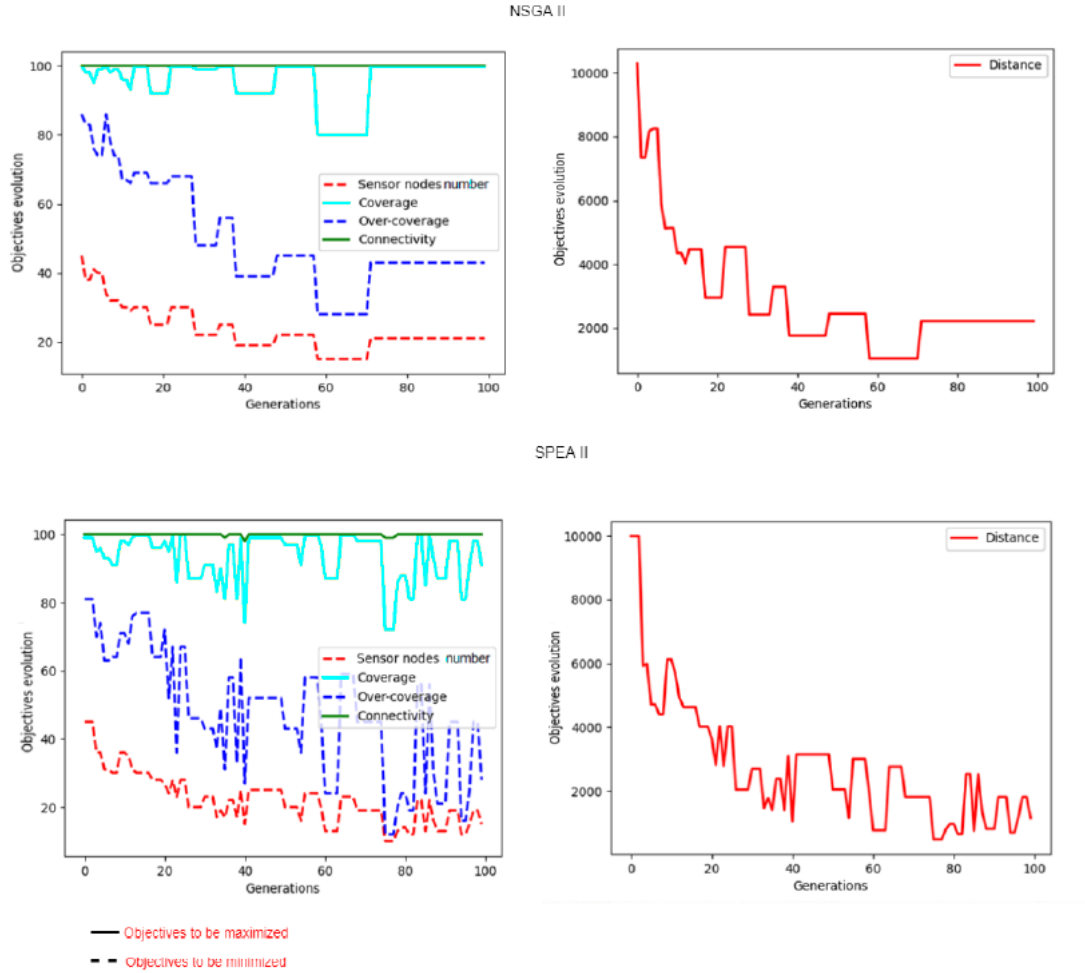


FIGURE 2.16 – Comparaison entre NSGA II et SPEA II en fonction de l'évolution de leurs fonctions objectives.

Temps d'exécution : Le NSGA II présente un avantage en termes de temps d'exécution avec une durée de 139.12 secondes, contre 150.54 secondes pour le SPEA II. Cette différence, bien que modeste, peut être significative dans des scénarios où le temps de calcul est crucial. Cela suggère que NSGA II pourrait être préféré dans des contextes où la rapidité de convergence est prioritaire.

Nombre de capteurs : En termes de nombre de capteurs utilisés, NSGA II déploie un pourcentage plus élevé de capteurs (21%) par rapport à SPEA II (15%). Cela pourrait indiquer que NSGA II tend à maximiser la couverture en utilisant davantage de capteurs, ce qui peut entraîner des coûts supplémentaires en ressources matérielles.

Couverture : La couverture est un critère essentiel dans le déploiement de WSN. NSGA II atteint une couverture complète de 100 %, tandis que SPEA II obtient une

couverture légèrement inférieure de 93 %. Cette performance maximale de NSGA II en matière de couverture le rend particulièrement adapté aux applications nécessitant une surveillance exhaustive sans zones non couvertes.

Sur-couverture : La sur-couverture, qui reflète la redondance dans le déploiement des capteurs, est plus élevée avec NSGA II (43 %) comparé à SPEA II (24 %). Une sur-couverture élevée peut entraîner une utilisation inefficace des ressources, avec des capteurs déployés inutilement dans des zones déjà couvertes. SPEA II, avec une sur-couverture plus faible, montre une meilleure gestion des ressources en évitant les redondances.

Connectivité : Les deux algorithmes parviennent à maintenir une connectivité parfaite (100 %), ce qui est essentiel pour garantir la fiabilité et la robustesse du réseau. Cette capacité à maintenir une connectivité optimale indique que les deux méthodes sont capables de garantir une communication efficace entre les capteurs.

Distance : La distance totale, qui mesure l'efficacité spatiale du déploiement des capteurs, est nettement inférieure pour SPEA II (1098.98 mètres) par rapport à NSGA II (2214.94 mètres). Une distance totale plus faible indique que SPEA II parvient à placer les capteurs de manière plus compacte et optimisée, réduisant ainsi les coûts de communication et d'énergie.

Somme pondérée des objectifs de fitness La somme pondérée des objectifs de fitness est une mesure globale de la performance des algorithmes. SPEA II obtient une meilleure valeur (-188.99) comparée à NSGA II (-415.78), ce qui suggère une performance supérieure en termes d'atteinte des objectifs multiples. Cela indique que SPEA II est globalement plus efficace pour équilibrer les différents objectifs d'optimisation.

TABLE 2.2 – Comparaison entre SPEA II et NSGA II en fonction de la qualité des solutions et du temps d'exécution (en secondes).

MOGAs	Temps d'exécution (s)	Nombre de capteurs (%)	Couverture (%)	Sur-couverture (%)	Connectivité (%)	Distance (m)
SPEA II	150,54	15,0	93,0	24,0	100,0	1 098,98
NSGA II	139,12	21,0	100,0	43,0	100,0	2 214,94

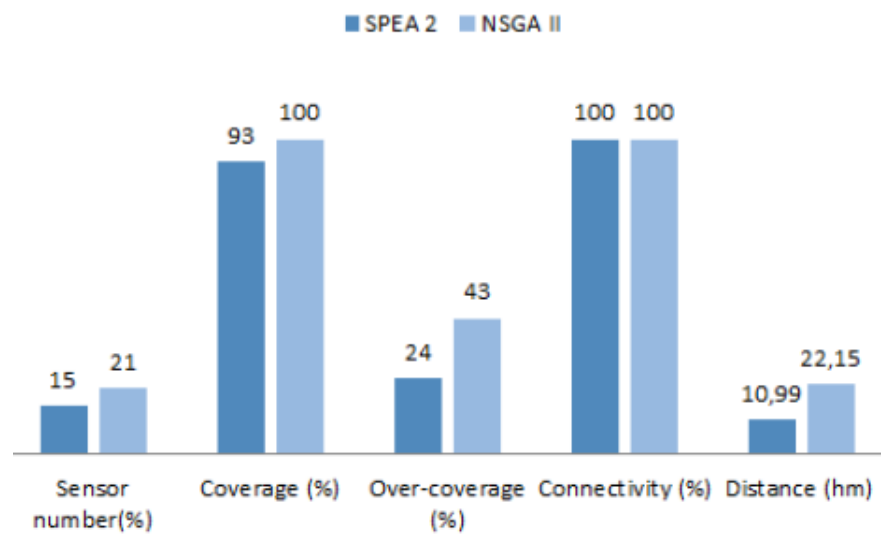


FIGURE 2.17 – Comparaison entre les solutions non dominées NSGA II et SPEA II selon leurs fonctions objectives.

Bien que NSGA II présente des avantages en termes de temps d'exécution et de couverture maximale, SPEA II démontre une meilleure optimisation spatiale avec une distance totale plus faible et une gestion plus efficace des ressources grâce à une sur-couverture réduite. La somme pondérée des objectifs de fitness supérieure pour SPEA II renforce sa position comme algorithme préférentiel pour les scénarios nécessitant un équilibre optimal entre les divers objectifs de déploiement de WSN. Ces résultats soulignent l'importance de choisir l'algorithme en fonction des priorités spécifiques de l'application, qu'il s'agisse de minimiser les coûts et la redondance, de maximiser la couverture ou de minimiser le temps d'exécution. La figure 2.18, montre le plan de déploiement de la solution non dominée de SPEA II.

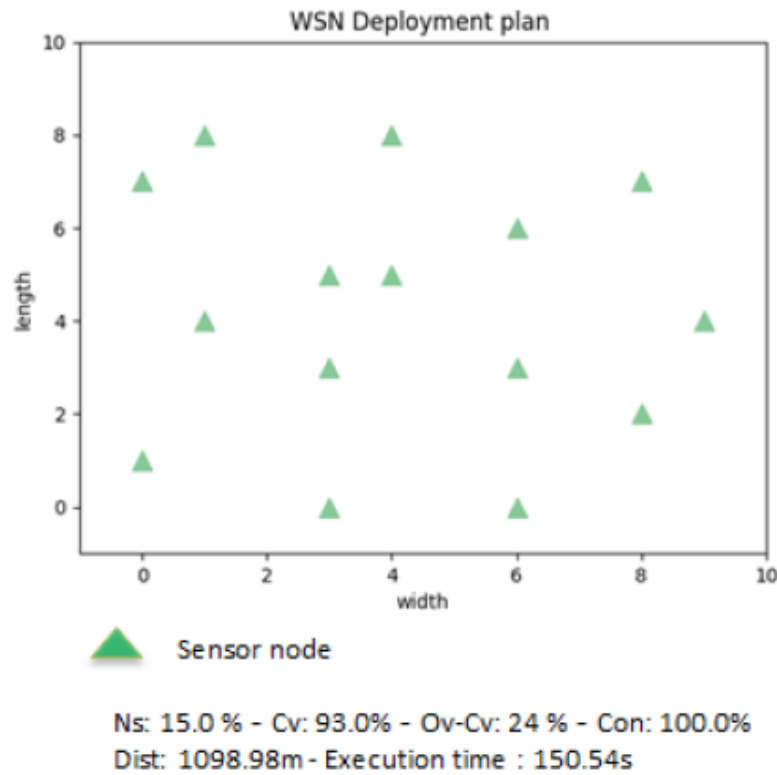


FIGURE 2.18 – Plan de déploiement du nœud capteur pour une parcelle agricole de 100 m^2 .

2.8.3 Limite du modèle proposé

Notre modèle proposé a prouvé ses performances dans le déploiement de nœuds pour une zone agricole selon les 5 contraintes décrites ci-dessus. Cependant, nous avons constaté qu'il montre une augmentation drastique du temps de calcul, lorsque nous augmentons la taille de la parcelle agricole comme indiqué dans le tableau 3 (pour 400 m^2 , le temps d'exécution était d'environ 40 min). Les résultats présentés dans le tableau 3 sont obtenus par l'exécution du modèle pour différentes parcelles de différentes tailles en utilisant le SPEA II.

Le phénomène d'augmentation du temps s'explique par l'augmentation de la taille des individus qui constituent la population car le temps d'exécution est proportionnel à la taille de la parcelle. Ainsi, la taille croissante des individus affecte le temps de calcul de l'évaluation de la fonction objective (Figure 18), qui est connue comme la phase la plus chronophage du processus MOEAs [234]. Et compte tenu de la qualité des solutions, il est essentiel que si la taille de la parcelle est grande, la taille de

TABLE 2.3 – La relation entre l'augmentation de la taille des parcelles, le temps d'exécution et la qualité des solutions pour les mêmes hyper-paramètres.

Dimensions de la parcelle (m^2)	Temps(s) d'exécution	Évaluation des fonctions objectifs (N_{sr} , C_{vr} , $O_v - C_{vr}$, C_{on} , Dist)
10×10	150,54 (2,50 minutes)	(15,0, 93,0, 24,0, 100,0, 1098,98)
10×15	387,72 (6,46 minutes)	(19,33, 99,33, 43,33, 100,0, 5520,67)
10×20	578,63 (9,64 minutes)	(22,5, 93,5, 54,0, 100,0, 15584,02)
15×20	601.33 (10.02 minutes)	(22,0, 94,0, 50,5, 99,5, 14793,52)
20×20	2342.59 (39,04 minutes)	(30,5, 99,75, 77,25, 100,0, 151859,30)

la population et le nombre de générations doivent augmenter de plus en plus afin de mieux explorer l'espace de recherche. Comme le montre le tableau 2.3, l'évaluation des fonctions objectives se détériore avec l'augmentation de la taille de la parcelle. Car dans nos tests, nous avons gardé les mêmes hyper-paramètres (taille de la population, nombre de générations) dans tous les tests. Ainsi, nous pouvons les comparer de manière équitable.

Par conséquent, pour confronter cette limite nous avons proposé des solutions pour minimiser le temps des évaluations en utilisant des solutions matérielles et logicielles pour accélérer les MOEAs. Afin de résoudre le problème des évaluations ils existent plusieurs méthodes comme présente le diagramme 2.19.

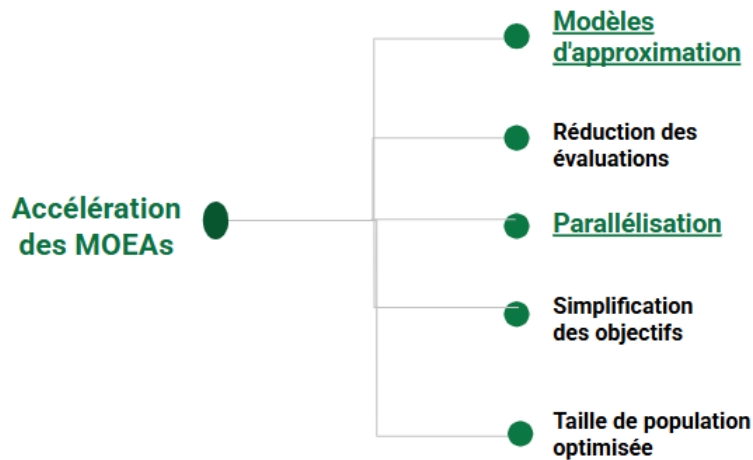


FIGURE 2.19 – Approches pour réduire le temps d'exécution des MOEAs.

Par rapport à la réduction de la taille de la population, dans notre cas cette méthode peut affecter la qualité des solutions obtenues. Pour la méthode de simplification des fonctions objectives, dans notre cas elles sont déjà très simplifiées et par rapport à la méthode de Réduction des évaluations elle peut causer une perte de diversité. De ce fait, pour l'accélération de notre modèle nous avons choisi les deux méthodes restantes.

La première c'est une méthode logicielle, pour minimiser le temps des évaluations nous avons utilisé une approximation des fonctions objectives par l'utilisation des modèles approximatifs en appliquant des algorithmes d'apprentissage automatique de régression. Concernant la deuxième méthode qui est une solution matérielle, nous avons proposé une architecture parallèle des MOEAs en se basant sur la nature de la parallélisation des MOEAs, qui peut être implémentée par la suite sur des plates-formes de calcul matérielles spécifiques telles que les unités de processeur graphique (Graphic Processing Unit, GPU).

2.9 Conclusion

L'étude approfondie des MOEAs a montré leur pertinence et leur efficacité pour résoudre le problème d'optimisation du déploiement des WSNs en agriculture. Les adaptations spécifiques des algorithmes MOEAs, telles que SPEA II et NSGA II, ont permis de mieux répondre aux exigences complexes et variées de ce problème. La comparaison entre SPEA II et NSGA II a révélé que, bien que les deux algorithmes

soient capables de fournir des solutions de haute qualité, ils présentent des différences en termes de convergence et de diversité des solutions.

Cependant, une limite significative du modèle proposé réside dans le temps de calcul élevé des MOEAs, particulièrement avec l'augmentation de la taille des parcelles agricoles. Cette complexité computationnelle peut rendre l'application de ces algorithmes impraticable pour des déploiements à grande échelle. Pour surmonter ce défi, il est essentiel de développer des solutions pour accélérer ces algorithmes. Dans le prochain chapitre, nous aborderons diverses stratégies d'accélération, telles que l'optimisation des algorithmes, l'utilisation de techniques de parallélisation et l'application de méthodes hybrides, pour rendre les MOEAs plus adaptés aux exigences de l'agriculture moderne à grande échelle.

Chapitre 3

Accélération des MOEAs

3.1 Introduction

Les MOEAs sont largement utilisés pour résoudre des problèmes complexes nécessitant l'optimisation simultanée de plusieurs objectifs conflictuels. Cependant, leur application pratique est souvent limitée par des exigences computationnelles élevées, en particulier lorsqu'il s'agit de problèmes à grande échelle ou de simulations coûteuses. Afin d'accélérer ces algorithmes, deux approches principales peuvent être envisagées : l'accélération logicielle et l'accélération matérielle.

L'accélération logicielle implique souvent l'hybridation des MOEAs avec d'autres algorithmes, tels que les algorithmes de régression, pour approximativement estimer les valeurs de fitness. Des méthodes comme les Perceptrons Multicouches (Multi-layer Perceptron, MLP), les k-plus Proches Voisins (K-Nearest Neighbors, KNN), les Arbres de Décision (Decision Tree, DT) et les Forêts d'Arbres Décisionnels (Random Forest, RF) peuvent être intégrées pour fournir des approximations rapides et précises, réduisant ainsi le nombre d'évaluations de fitness nécessaires.

D'autre part, l'accélération matérielle utilise des technologies avancées de calcul parallèle, telles que les unités de traitement graphique (Graphic Processing unit, GPU), pour répartir les calculs de fitness sur plusieurs cœurs de traitement, augmentant considérablement la vitesse de traitement des MOEAs.

Ce chapitre explore ces deux approches d'accélération, en mettant particulièrement l'accent sur l'efficacité de l'hybridation logicielle avec des algorithmes de régression et la parallélisation des calculs via GPU.

Partie 1 :

Accélération logicielle des MOEAs : Approximation de la fonction fitness

3.2 Hybridation des méta-heuristiques

Afin d'améliorer la capacité de recherche des algorithmes méta heuristiques, l'hybridation qui est la corporation entre deux ou plusieurs méthodes ou bien algorithmes, vise à combiner les avantages de chaque algorithme pour former un algorithme hybride, tout en essayant simultanément de corriger les inconvénients substantiels. Dans les cas généraux, les algorithmes hybrides peuvent être meilleurs en termes de compétences informatiques et de vitesse, ou en précision. Autrement dit, l'hybridation consiste à combiner les caractéristiques de deux méthodes différentes pour tirer les avantages des deux méthodes [166]. Dans le cas de l'optimisation, l'hybridation des algorithmes méta heuristique a permis l'ouverture de nouveau champs dans le domaine de l'optimisation combinatoire.

Les origines des algorithmes hybrides des méta-heuristiques reviennent aux travaux de Glover [166], J. J. Grefenstette [191] et Mühlenbein et al [235]. Chacun d'eux a introduit une méthode de descente simple pour améliorer une recherche évolutive. Mais à cette période, la plupart des chercheurs n'y accordaient que peu d'intérêts. Actuellement, les méta-heuristiques hybrides sont devenues plus populaires car les meilleurs résultats trouvés pour plusieurs problèmes d'optimisation combinatoires ont été obtenus avec des algorithmes hybrides. L'hybridation des méta-heuristiques peut être divisée en deux grandes parties : hybridation des méta-heuristiques avec des méta-heuristiques et hybridation des méta-heuristiques avec des méthodes exactes.

Selon les travaux effectués dans [166] et de [236], selon le type de la problématique traitée les algorithmes hybrides peuvent être classés en deux catégories : problèmes de conception et problèmes d'implémentation. La première catégorie concerne l'algorithme en lui-même, incluant les problèmes de fonctionnalité et d'architecture de l'algorithme hybride développé. La seconde, concerne la plate-forme matérielle utilisée, les modèles de programmation, ainsi que, l'environnement dans lequel l'algorithme est exécuté.

3.3 Problèmes de conception

Selon [166], il existe deux types de classification dans les problèmes de conception. La première concerne la classification hiérarchique qui elle-même contient l'hybridation à faible niveau (ou bas-niveau) et hybridation à haut-niveau. La deuxième classification est liée au choix des caractéristiques qui peut être traitée indépendamment du choix de conception précédent. Dans ce cas là, la classification devient horizontale (plate).

3.3.1 Classification hiérarchique des méta-heuristiques

L'hybridation hiérarchique des méta-heuristiques permet d'exploiter les forces complémentaires de différents algorithmes, améliorant ainsi l'efficacité et la qualité des solutions pour des problèmes de conception complexes. En décomposant le problème en sous-problèmes et en coordonnant les efforts d'optimisation, cette approche offre une robustesse et une adaptabilité accrues, adaptées à divers domaines d'application.

Cette classification comme le montre la Figure 3.1 est caractérisée par le niveau d'intégration et le mode de l'hybridation. Le niveau d'hybridation peut être bas (Low-Level) ou haut (High-Level). Dans le niveau bas, une méta-heuristique remplace un opérateur d'une autre méthode qui l'englobe. Par contre, dans le niveau haut de l'hybridation, chaque méta-heuristique garde sa propriété au cours de l'hybridation. Chaque niveau d'hybridation engendre deux modes de coopération à savoir, le mode relais et mode co-évolutionnaire. Dans le mode relais, les méthodes sont exécutées séquentiellement, c'est-à-dire le résultat de la première méthode est le début de la méthode suivante. Quand les différentes méthodes fonctionnent en parallèle pour explorer l'espace de recherche, on parle de mode co-évolutionnaire. La combinaison des modes et des niveaux donne quatre classes d'hybridation qui sont l'hybridation relais de bas niveau, l'hybridation co-évolutionnaire de bas niveau, l'hybridation relais de haut niveau et l'hybridation co-évolutionnaire de haut niveau[237].

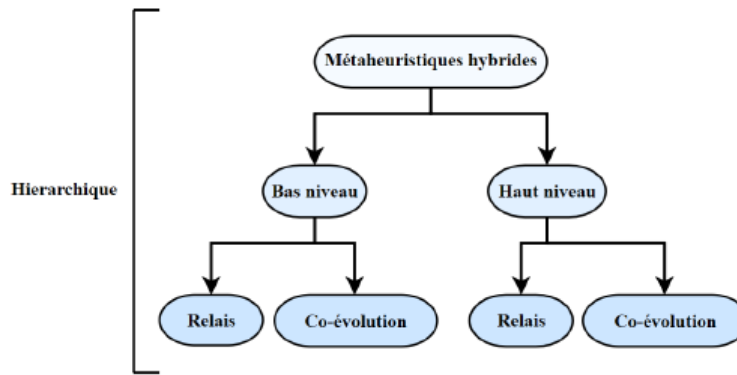


FIGURE 3.1 – Schéma de classification hiérarchique des méta-heuristiques hybrides.

3.3.1.1 Hybridation relais de bas niveau

L'hybridation de relais bas niveau (Low-Level Relay Hybridization) est une stratégie où les différentes méta-heuristiques sont intégrées les uns après les autres de manière séquentielle, avec une méta-heuristique prenant le relais d'une autre pour continuer à améliorer les solutions trouvées. Ce type d'hybridation est caractérisé par une étroite collaboration entre les algorithmes, où chaque algorithme utilise les résultats intermédiaires de l'autre pour progresser. En prend par exemple, [238] ou l'hybridation du PSO avec le recuit simulé a été appliqué en sorte que le PSO est utilisé pour une exploration globale de l'espace de solutions et le SA affine les solutions les plus prometteuses pour un ajustement local et pour échapper aux minima locaux. Dans [239], afin de résoudre le problème de conception de circuits électroniques où des solutions initiales doivent être explorées globalement et ensuite optimisées pour réduire les interférences et minimiser la consommation d'énergie. Le GA a été utilisé pour générer des configurations de circuits en utilisant des opérateurs de sélection, de croisement et de mutation et l'algorithme de recherche tabou affine les configurations pour améliorer la performance en termes de réduction des interférences et de minimisation de la consommation d'énergie.

3.3.1.2 Hybridation co-évolutionnaire de bas niveau

(Low-level Teamwork Hybrid) consiste à incorporer une ou plusieurs méta-heuristiques à base de solution unique dans une méta-heuristique à population de

solutions. L'avantage de ce type d'hybridation est de compenser la puissance d'exploitation d'une recherche locale et celle d'exploration d'une recherche globale. Stützle et Hoos [240] incorporent une fonction de recherche locale dans un algorithme de colonie de fourmis pour résoudre le problème du voyageur de commerce et celui de partition de graphes. Gambardella, Taillard et al. [241] ont utilisé un algorithme de descente dans un algorithme de colonie de fourmis pour résoudre le problème de partition de graphes et McKendall et Shang [242] ont amélioré la méthode proposée par Gambardella, Taillard et al, en remplaçant l'algorithme de descente par l'algorithme du recuit simulé. Azimi [243] dans sa recherche de résolution du problème d'agencement d'horaire d'examens a proposé une amélioration de la solution de colonie de fourmis pour chaque cycle en appliquant le recuit simulé. Dans un algorithme génétique, J. Suh et D. van Gucht [244] ont remplacé l'opérateur de mutation par une recherche de descente et C. Fleurent et Ferland [245] ont appliqué cette classe d'hybridation pour résoudre le problème de partition de graphe en remplaçant l'opérateur de mutation d'un algorithme génétique par une recherche tabou.

3.3.1.3 Hybridation relais de haut niveau (High-level Relay Hybrid)

Les méta-heuristiques sont utilisées de manière séquentielle c'est-à-dire la (ou les) solution(s) finale(s) de la première méta-heuristique est la (ou les) solution (s) initiale(s) de la méta-heuristique suivante. Dans cette procédure, toutes les méthodes gardent leur intégrité. Talbi et al. [246] ont introduit la recherche taboue à la fin d'un algorithme génétique pour améliorer les solutions obtenues. Comme il est connu d'utiliser une recherche locale pour initialiser une autre méta-heuristique, Azimi a utilisé deux hybridations de type relais de haut niveau pour résoudre le problème d'agencement d'horaires d'examens.

Premièrement, il a initialisé les phéromones des colonies de fourmis par la recherche taboue puis il a pris la solution trouvée par les algorithmes de colonie de fourmis comme une solution initiale d'une recherche taboue. Pour résoudre le problème d'assignation quadratique, Lin, Kao et al, dans [247], proposent une hybridation où la méthode du recuit simulé crée une population initiale pour un GA.

3.3.1.4 Hybridation co-évolutionnaire de haut niveau (High-level Team-work Hybrid)

Les méta-heuristiques utilisées travaillent en parallèle en échangeant des informations entre elles afin de trouver la solution optimale du problème posé. L'exemple le plus populaire est celui de l'algorithme génétique basé sur le modèle insulaire proposé par Tanese [248]. Pour cette hybridation, la population est divisée en sous-populations réparties sur les sommets d'un hypercube dont lesquels un algorithme génétique est lancé. Chaque sommet correspond à une zone de recherche de solutions et périodiquement, des individus migrent entre les sommets en contribuant à trouver les solutions optimales.

Dans [249], Ghédira et Hammami ont fait appel à cette hybridation pour résoudre le problème de partitionnement de graphe. Ils ont utilisé la recherche tabou 2 et le recuit simulé de telle sorte qu'à chaque intervalle régulier, les deux méthodes échangent d'informations pour trouver la meilleure solution. Notons qu'il existe d'autres travaux basés sur cette hybridation en utilisant des recuits simulés [250], des recherches tabou [166], la programmation génétique [251] et la stratégie évolutionnaire [252]. Dans tous ces travaux, il a été démontré [253] que les résultats ne sont plus performants si les méta-heuristiques n'échangent pas d'informations.

3.3.2 Classification à plat des méta-heuristiques

La classification à plat des méta-heuristiques comme le montre la Figure 3.2 est caractérisée par le type des méthodes hybridées, leur domaine d'application et la nature de leurs fonctions. Selon le type d'hybridation, on trouve des méthodes hybridées homogènes où les algorithmes utilisés se basent sur la même méta-heuristique comme le modèle insulaire et des méthodes hybridées hétérogènes où les méta-heuristiques utilisées sont différentes. Le modèle proposé dans [249] est une hybridation de haut niveau co-évolutionnaire hétérogène. Le domaine d'application des méta-heuristiques hybridées permet de distinguer deux grandes classes d'hybridation, les hybridations globales et les hybridations partielles. L'hybridation globale a lieu lorsque toutes les méthodes hybridées sont appliquées à la totalité de l'espace de recherche. Toutes les méthodes que nous avons étudiées précédemment sont des hybridations globales. A

l’opposé, l’hybridation partielle décompose un problème en sous-problèmes où chacun a son propre espace de recherche.

Dans ce contexte, Taillard [254] propose une décomposition du problème de routage de véhicules en appliquant la recherche tabou. L’idée est de diviser l’ensemble des villes à visiter en secteurs indépendants dont chacun représente un espace de recherche. Dans un autre article, Taillard et Voss [255] ont suggéré une méthode pour créer des sous problèmes à partir du problème initial et le résoudre à l’aide d’une méta-heuristique ou d’une méthode exacte. Selon le problème traité, Talbi distingue deux types d’hybridation, une hybridation généraliste et une hybridation spécialiste. On parle d’hybridation généraliste quand toutes les méta-heuristiques hybridées traitent le même problème d’optimisation. Toutes les hybridations que nous avons citées dans la classification hiérarchique font partie de cette catégorie.

A l’inverse, les hybridations spécialisées se produisent lorsque chaque méta-heuristique traite un problème différent. Un exemple de ce type est l’utilisation d’une méta-heuristique pour initialiser les paramètres d’une autre méta-heuristique. Krueger [256] optimise les paramètres d’un recuit simulé à l’aide d’un algorithme génétique. Abbattista [257] optimise ceux d’un algorithme de colonie de fourmis à l’aide d’un AG et Shahookar et Mazumder [258] optimisent les paramètres d’un GA à l’aide d’un autre GA.

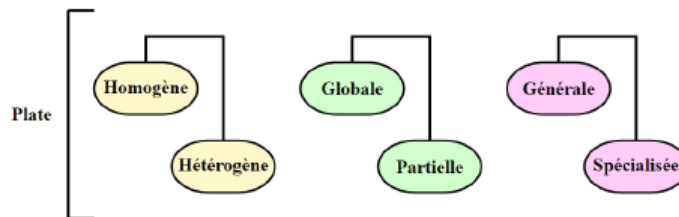


FIGURE 3.2 – Schéma de classification horizontale (plate) des méta-heuristiques hybrides.

Dans notre cas d’étude, le problème que nous allons traiter est un problème de conception, nous nous intéressons aux hybridation bas-niveau relais, où une technique (approximation) prend le relais pour une tâche spécifique (évaluation des fitness), donc l’algorithme hybride développé consiste en SPEA II utilisé dans le deuxième chapitre et une méta-heuristique qui va remplacer la fonction d’évaluation qui consomme trop

de ressources et temps de calcul sur tout quand il s'agit des problèmes de grande dimensions, ou le temps d'exécution du MOEAs augmentent exponentiellement en fonction des dimensions de la surface du déploiement.

3.4 Approximation de la valeur d'évaluation

L'approximation des valeurs de fitness dans le contexte des méta-heuristiques implique l'utilisation de méthodes visant à estimer la qualité des solutions candidates sans avoir besoin d'évaluer complètement la fonction de fitness. Cette approche est souvent utilisée pour accélérer le processus d'optimisation dans les méta-heuristiques, où l'évaluation précise de la fitness peut être coûteuse en termes de temps de calcul ou de ressources. Il existe aussi plusieurs situations dans lesquelles l'évaluation des candidats devient très difficile ou même impossible à exprimer analytiquement. Il est donc envisageable d'avoir une approximation efficace en minimisant le nombre d'évaluations de fitness afin de trouver une solution acceptable. Pour pallier ce problème, des modèles efficaces en termes de calcul peuvent être intégrés pour approximer la fonction objective. Ces modèles sont souvent appelés modèles approximatifs ou méta modèles. Voici quelques situations courantes où l'approximation des valeurs de fitness est bénéfique :

- **Problèmes à Grande Échelle** : Lorsque les problèmes à résoudre sont de grande taille, avec un grand nombre de variables ou de contraintes, l'évaluation précise de la fonction de fitness pour chaque solution peut être extrêmement coûteuse. L'approximation permet de réduire la charge computationnelle associée à ces problèmes.
- **Optimisation Intensive** : Dans les applications nécessitant une optimisation intensive, telles que la conception de réseaux de communication ou la planification logistique, l'approximation des valeurs de fitness permet d'accélérer le processus d'optimisation sans compromettre la qualité des solutions.
- **Données Complexes** : Lorsque les données à évaluer sont complexes, bruitées ou incomplètes, l'utilisation de modèles d'approximation peut aider à obtenir des estimations fiables de la fitness en évitant le surentraînement sur des données non représentatives.

- **Exploration d'Espaces de Solutions Étendus** : Dans le cadre de l'exploration d'espaces de solutions étendus, tels que la recherche de nouvelles molécules en chimie ou l'optimisation de structures en ingénierie, l'approximation des valeurs de fitness permet de réduire le temps nécessaire à l'évaluation des candidats.
- **Optimisation Parallèle** : Lors de l'optimisation parallèle, où plusieurs solutions sont évaluées en même temps, l'approximation des valeurs de fitness permet d'accélérer l'évaluation des solutions sur chaque thread ou chaque processeur, améliorant ainsi l'efficacité globale de l'algorithme.
- **Optimisation en Ligne ou en Temps Réel** : Dans les systèmes nécessitant une optimisation en temps réel, comme les systèmes de recommandation en ligne ou les contrôleurs de processus industriels, l'approximation des valeurs de fitness permet d'obtenir rapidement des solutions satisfaisantes sans compromettre la performance du système.

L'objectif principal de l'approximation des valeurs de fitness est de permettre aux modèles approximatifs de substituer intégralement la fonction objective d'origine. Cependant, selon les conclusions de l'article [259], les chercheurs ont observé qu'il est souvent indispensable de fusionner le modèle approximatif avec la fonction objective originale afin d'assurer une convergence appropriée des MOEAs. Dans cette optique, la réévaluation de certains individus à l'aide de la fonction d'adéquation initiale, également connue sous le nom de contrôle de l'évolution, se révèle cruciale.

Du fait de la rareté des données disponibles pour l'apprentissage des modèles approximatifs et de la complexité élevée de l'espace d'entrée, parvenir à une approximation fonctionnelle globale de la fonction d'évaluation d'origine représente une tâche extrêmement ardue. Pour surmonter cet obstacle, deux approches principales peuvent être envisagées :

- Dans un premier temps, il est impératif d'utiliser le modèle approximatif en tandem avec la fonction d'évaluation d'origine, malgré sa disponibilité générale mais sa demande en ressources de calcul. Il est donc essentiel de l'exploiter de manière efficiente, ce qui correspond au concept de contrôle de l'évolution dans le domaine de l'optimisation évolutionnaire.

- Ensuite, il est crucial d'améliorer autant que possible la qualité du modèle approximatif, compte tenu des limitations liées au nombre restreint de données. Plusieurs aspects revêtent une importance capitale pour cette amélioration, notamment le choix adéquat du modèle, la sélection appropriée de la méthode d'apprentissage et la détermination des mesures d'erreur les plus pertinentes.

3.4.1 Méthodes d'approximation

L'approximation des valeurs de fitness est une technique cruciale pour améliorer l'efficacité des Algorithmes d'Évolution Multi-Objectifs (MOEAs), surtout lorsque les évaluations de fitness sont coûteuses en temps ou en ressources. Voici quelques méthodes courantes pour approximer les valeurs de fitness :

3.4.1.1 Modèles analytiques

La valeur fitness est estimée en utilisant des formules mathématiques ou de modèles statistiques en fonction des caractéristiques de la solution. Par exemple, dans le cas de l'optimisation de réseaux, des modèles analytiques peuvent estimer la qualité d'une configuration réseau en fonction de paramètres tels que la latence, la bande passante et la fiabilité [260].

3.4.1.2 Échantillonnage Aléatoire

Dans cette méthode, l'évaluation de la fitness sur un sous-ensemble aléatoire des données ou des variables de la solution. Par exemple, dans les algorithmes génétiques, une évaluation partielle de la fitness peut être effectuée sur un échantillon aléatoire des gènes pour estimer la qualité de la solution[261].

3.4.1.3 Méthodes d'Extrapolation

Utilisation des données obtenues lors d'évaluations précédentes pour extrapoler la fitness pour de nouvelles solutions similaires. Par exemple, dans les algorithmes évolutionnaires, les performances des solutions précédentes peuvent être utilisées pour estimer la qualité des solutions actuelles[262].

3.4.1.4 Apprentissage Machine

Utilisation de techniques d'apprentissage automatique pour construire un modèle prédictif de la fitness à partir des données d'entraînement. Par exemple, des réseaux de neurones peuvent être utilisés pour prédire la qualité des solutions à partir de caractéristiques spécifiques de la solution [263].

3.4.2 Types d'approximation

Chacune des méthodes d'approximation peut être adaptée en fonction du problème spécifique à résoudre et des contraintes de ressources disponibles. En combinant différentes méthodes ou en les adaptant à des contextes spécifiques, il est possible d'obtenir des approximations efficaces des valeurs de fitness dans les méta-heuristiques. Le concept d'approximation en optimisation n'est pas nouveau. Traditionnellement, il existe deux approches de base, l'approximation fonctionnelle et l'approximation du problème. En outre, des techniques d'approximation spéciales pour l'évaluation de fitness ont également été proposées [259].

3.4.2.1 Approximation du problème

Dans cette approche, les aspects complexes du problème sont simplifiés tout en préservant les caractéristiques essentielles qui permettent de trouver une solution acceptable. Cette approche est souvent utilisée dans des situations où la résolution exacte du problème est difficile ou coûteuse, notamment en termes de temps de calcul ou de ressources.

3.4.2.2 Approximation fonctionnelle

Dans l'approximation fonctionnelle, la nouvelle fonction doit conserver les propriétés importantes de la fonction originale, telles que les valeurs aux points clés, les tendances générales et les comportements asymptotiques. L'objectif principal de cette approche est de simplifier une fonction complexe en une forme plus simple et plus facile à manipuler, tout en conservant sa précision dans une certaine mesure. Cette technique permet de simplifier des fonctions complexes tout en réduisant la complexité des calculs, ce qui facilite l'interprétation des données.

3.4.2.3 Approximation évolutive

L'approximation évolutive est une approche qui combine des techniques d'approximation avec des méthodes évolutives telles que les algorithmes génétiques, les stratégies évolutionnaires ou d'autres méta-heuristiques. Cette approche vise à estimer ou à optimiser une fonction complexe en utilisant des méthodes évolutives pour ajuster progressivement les paramètres d'une fonction d'approximation. Elle se divise en deux classes d'algorithmes :

— **Héritage de la forme physique**

Dans ce processus, les évaluations des individus sont dérivées de la phase de reproduction et sont héritées des valeurs de forme de leurs parents. Par exemple, la valeur de fitness d'un enfant peut être estimée par une somme pondérée des valeurs de forme de ses parents. Cela signifie que lors de la reproduction, les caractéristiques des parents sont combinées de manière pondérée pour estimer la valeur de forme de leur enfant. Cette estimation peut être basée sur diverses techniques, telles que des méthodes d'interpolation, des modèles statistiques ou des algorithmes d'apprentissage automatique. Par exemple, si les parents ont des valeurs de forme (ou des valeurs de fitness) A et B, alors la valeur de forme de leur enfant peut être calculée comme une combinaison pondérée de ces valeurs, par exemple : Valeur de forme de l'enfant = α Valeur de forme de A + (1 - α) Valeur de forme de B. Où α est un coefficient de pondération qui peut être ajusté en fonction des caractéristiques spécifiques du problème et des préférences de l'utilisateur. Ce processus permet de conserver les caractéristiques importantes des parents tout en introduisant une variation dans la population. Il peut être utilisé dans les algorithmes évolutifs pour générer de nouvelles solutions en combinant les caractéristiques des solutions existantes de manière pondérée.

— **Imitation de la forme physique**

Dans ce processus, les individus sont d'abord classés en plusieurs groupes à l'aide d'une méthode de classification. Ensuite, seul un individu représentant chaque groupe est sélectionné pour être évalué par la fonction de fitness. Les valeurs des autres individus dans le même groupe sont ensuite estimées à partir

de cet individu représentatif, en se basant sur une mesure de distance. Plusieurs modèles d'approximation sont utilisés pour cette estimation :

- Réseaux de Neurones : Les réseaux de neurones sont des modèles flexibles capables d'apprendre des relations complexes entre les données. Ils sont utilisés pour estimer les valeurs des individus à partir de l'individu représentatif.
- Régression et Interpolation : Les techniques de régression et d'interpolation sont utilisées pour estimer les valeurs manquantes en fonction des valeurs connues. Elles sont adaptées pour estimer les valeurs des individus sur la base de l'individu représentatif et des mesures de distance.
- Modèle de Krigeage : Le modèle de Krigeage, également connu sous le nom de krigeage spatial, est utilisé pour estimer les valeurs dans des domaines spatiaux. Il est particulièrement utile lorsque les données sont corrélées spatialement et est souvent utilisé pour estimer les valeurs des individus à partir de l'individu représentatif et des mesures de distance.

Ces modèles d'approximation sont sélectionnés en fonction des caractéristiques des données et des exigences spécifiques de l'application. Ils permettent de prédire les valeurs des individus non évalués de manière efficace et précise, ce qui améliore l'efficacité globale de l'algorithme évolutif.

3.4.3 Incorporation de modèles de fitness approximatifs

L'intégration de modèles d'approximation pour les évaluations de fitness peut considérablement réduire le nombre d'évaluations nécessaires, comme démontré dans [264]. Cependant, l'application de ces modèles dans le cadre du calcul évolutionnaire est complexe. Un aspect crucial est la difficulté de construire un modèle d'approximation globalement précis, en raison des limitations en nombre et qualité des données d'apprentissage disponibles. Il est observé que l'utilisation d'un modèle d'approximation pour les évaluations de fitness peut entraîner une convergence vers un faux optimum. Ce faux optimum correspond à un optimum du modèle approximatif, qui ne correspond pas à celui de la fonction de fitness originale, comme illustré dans la Figure 3.3.

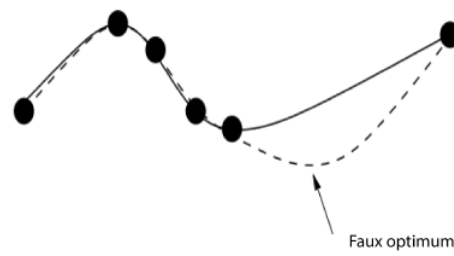


FIGURE 3.3 – Faux Optimum d'un modèle approximatif.

Ainsi, dans la plupart des situations, il est vital d'employer le modèle approximatif en tandem avec la fonction de fitness originale, ce qui caractérise le processus d'évolution comme contrôlé.

3.4.3.1 Contrôle d'évolution

Le contrôle de l'évolution dans l'approximation des fitness se réfère à l'utilisation simultanée de modèles approximatifs et de fonctions de fitness originales pour évaluer la pertinence des individus dans le cadre de méthodes évolutives (contrôle d'évolution dans [265]). Cette approche vise à combiner les avantages des modèles approximatifs, qui réduisent le nombre d'évaluations de fitness nécessaires, avec la précision offerte par les fonctions de fitness originales. En contrôlant l'évolution de cette manière, il est possible d'obtenir des solutions efficaces tout en évitant les risques associés à la convergence vers des optimaux erronés induits par l'utilisation exclusive de modèles approximatifs. Il existe généralement deux types de contrôle de l'évolution, l'un basé sur l'individu, l'autre sur la génération :

- **Contrôle de l'évolution individuel** : Dans ce type, l'individu qui est évalué à l'aide de la fonction de fitness originale est appelé un individu contrôlé et chaque individu est évalué en fonction de son propre modèle approximatif, qui est ensuite comparé à la valeur de fitness réelle obtenue à partir de la fonction de fitness originale. Cela permet de déterminer si l'individu a convergé vers une solution adéquate ou s'il nécessite une évaluation plus précise.
- **Contrôle de l'évolution générationnel** Ce type de contrôle évalue l'ensemble de la population à chaque génération en utilisant un modèle approximatif global. Les individus sont classés en fonction de leurs valeurs de fitness estimées, et seuls

certaines individus sélectionnés sont évalués à l'aide de la fonction de fitness originale. Cette approche permet de réduire le nombre d'évaluations nécessaires tout en maintenant une certaine précision dans la sélection des meilleurs individus.

Ces deux types de contrôle de l'évolution sont utilisés pour adapter l'utilisation des modèles approximatifs dans le cadre du calcul évolutionnaire, en tenant compte à la fois de l'efficacité et de la précision des évaluations de fitness.

3.4.4 Apprentissage automatique supervisé

L'apprentissage automatique supervisé (Supervised Machine Learning, SML) est une approche puissante pour l'approximation des valeurs de fitness dans divers domaines, tels que l'optimisation, la simulation et l'analyse de données. Étant une branche de l'intelligence artificielle, l'apprentissage automatique supervisé est formé sur un ensemble de données étiquetées. Ces données étiquetées consistent en paires d'entrées et de sorties désirées, permettant à l'algorithme d'apprendre à mapper les entrées vers les sorties attendues. Le terme « apprentissage automatique » fait généralement référence à un programme informatique qui reçoit des entrées et produit des sorties [265].

Pendant l'apprentissage, le modèle établit une relation entre les données d'entrée et de sortie en ajustant ses paramètres internes. Une méthode courante d'entraînement consiste à fournir à chaque itération un ensemble d'entrées associées à leurs sorties correspondantes, permettant ainsi au modèle de se régler progressivement pour produire les sorties désirées. Ce processus d'ajustement des paramètres constitue l'essence de l'apprentissage. Les deux principaux types d'apprentissage supervisé sont la classification et la régression, qui seront abordés dans les sections suivantes. Chaque modèle supervisé est caractérisé par un ensemble de paramètres internes ajustables et un algorithme qui les règle afin d'atteindre les objectifs fixés.

3.4.5 Approximation des valeurs de fitness par SML

Les modèles générés par les algorithmes de SML peuvent être entraînés à partir de données d'entraînement contenant à la fois des caractéristiques d'entrée et les valeurs de fitness correspondantes. Une fois entraînés, ils peuvent être utilisés pour estimer les

valeurs de fitness pour de nouvelles observations, ce qui est utile dans de nombreuses applications, notamment l'optimisation, la prédiction et la simulation. Dans la suite nous allons présenter quelques algorithmes couramment utilisés.

3.4.5.1 Régression linéaire

La régression linéaire est une technique statistique fondamentale utilisée pour modéliser la relation entre une variable dépendante et une ou plusieurs variables indépendantes en ajustant une ligne droite aux données. Cependant, cette méthode peut être sujette à une lenteur accrue, avec un nombre d'itérations fortement dépendant de l'échelle du problème.

Malheureusement, La régression linéaire suppose une relation linéaire entre les variables d'entrée et la valeur de fitness, ce qui peut ne pas être le cas pour notre problème d'optimisation ou la relation entre les différents types de déploiement et leurs valeurs de fitness ne peuvent pas être linéaire.

3.4.5.2 Réseaux de neurones (ANN)

Les réseaux de neurones Artificiels (Artificial Neural Network, ANN) [266] dominent la sphère de la classification, de l'estimation, de la prédiction et de la segmentation. Fondés sur des modèles biologiques, ils sont composés de neurones organisés selon une architecture spécifique. Leur optimisation s'opère généralement par le biais de méthodes d'apprentissage statistique, favorisant des décisions basées sur la perception plutôt que sur un raisonnement logique formel. Un ANN artificiel est constitué de multiples neurones, reliés par des liens leur permettant d'envoyer et de recevoir des signaux. Chaque connexion se voit attribuer un poids, influençant ainsi le neurone connecté. Par conséquent, chaque neurone possède une entrée pour recevoir des informations, peut avoir une fonction d'activation, et enfin, générer une sortie.

3.4.5.3 k-plus proches voisins (KNN)

L'algorithme des k-plus proches voisins est une méthode non paramétrique utilisée pour la classification et la régression. Dans les problèmes de régression, l'algorithme kNN est utilisé pour estimer les valeurs continues en fonction des valeurs des k voisins

les plus proches dans l'espace des caractéristiques. Cet algorithme commence par stocker toutes les données de formation, y compris les caractéristiques (features) et les valeurs correspondantes, et pour chaque point de test, l'algorithme calcule la distance entre le point de test et tous les points de formation. La distance peut être calculée de différentes manières, mais la distance euclidienne est la plus courante[267].

L'algorithme k-NN est une méthode puissante et flexible pour l'approximation des valeurs dans des problèmes complexes, offrant une approche intuitive pour la régression non paramétrique. Bien que le calcul puisse être intensif, l'algorithme reste une technique populaire en raison de sa simplicité et de sa capacité à traiter des données sans nécessiter de modèles paramétriques stricts.

3.4.5.4 Arbres de décisions (DT)

L'algorithme DT se présente sous la forme d'un diagramme ou d'un graphique facilitant l'établissement d'un plan d'action ou la représentation d'une probabilité statistique. Son nom découle de sa ressemblance avec un arbre, généralement représenté comme un diagramme vertical ou horizontal se ramifiant. Chaque "branche" de l'arbre de décision représente une décision, un résultat ou une réaction possible à partir de la décision initiale. Les branches terminales, ou "feuilles", marquent les résultats finaux de chaque chemin de décision. Les arbres de décision sont souvent utilisés comme modèles de régression dans divers domaines d'optimisation, également connus sous le nom de diagrammes en arbre en mathématiques. Un ensemble d'arbres de décision peut générer des forêts aléatoires, ces modèles peuvent capturer des relations complexes entre les caractéristiques et la valeur de fitness. Leur robustesse et leur capacité à gérer des données hétérogènes en font des choix populaires[268].

Cependant, ces algorithmes restent très sensibles aux données spécifiques sur lesquelles ils sont formés. Si les données d'entraînement sont modifiées, l'arbre décisionnel résultant peut être très différent et, à son tour, les prédictions peuvent être très différentes. De plus, les arbres décisionnels sont coûteux à former, présentent un risque élevé de sur ajustement et ont tendance à trouver des optima locaux, car ils ne peuvent pas revenir en arrière après avoir effectué une division.

Pour surmonter certaines de ces limites, des approches comme l'élagage (pruning) pour réduire le sur-ajustement, l'utilisation de forêts aléatoires ou de méthodes d'ensemble (comme le boosting), et l'intégration de validations croisées peuvent être utilisées. Ces techniques peuvent améliorer la robustesse, la précision et la généralisation des arbres de décision pour l'approximation des valeurs de fitness. Pour remédier à ces faiblesses, nous nous tournons vers les méthodes d'ensemble, qui illustrent la puissance de la combinaison de plusieurs modèles (ex : arbre de décision) en un seul modèle. Une méthode d'ensemble est une technique qui combine les prédictions de plusieurs algorithmes d'apprentissage automatique et d'obtenir des prédictions plus précises que tout modèle individuel. Un modèle composé de plusieurs modèles est appelé et un modèle d'ensemble. Les méthodes d'ensemble les plus connues sont "le bagging", également connues sous le nom d'agrégation bootstrap, ou "le boosting".

- **Boosting** : Le boosting (Figure 3.4) construit un ensemble de modèles séquentiellement, où chaque modèle tente de corriger les erreurs de son prédécesseur. Contrairement au bagging, qui forme des modèles indépendants, le boosting ajuste chaque modèle en fonction des performances des précédents. Les modèles sont pondérés en fonction de leur précision, et plus de poids est donné aux observations mal classées. AdaBoost (Adaptive Boosting) et Gradient Boosting Machines (GBM) sont des exemples populaires de méthodes de boosting [269]. Ces techniques sont particulièrement efficaces pour réduire le biais et améliorer la précision des modèles.

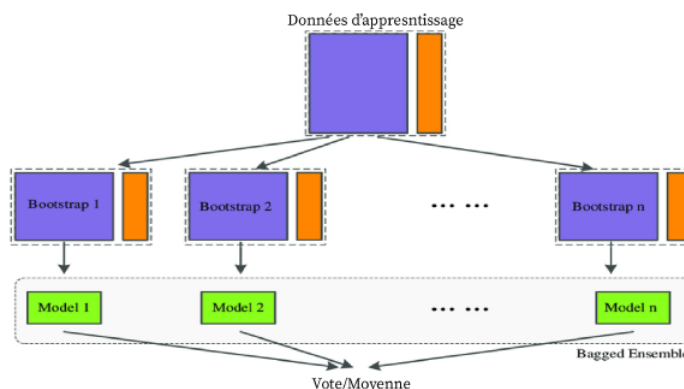


FIGURE 3.4 – Méthode d'agrégation Bootstrap [270].

- **Aggregation Bootstrap (Bagging)** : Cette méthode consiste à entraîner plusieurs modèles indépendants (comme des arbres de décision) sur des sous-ensembles aléatoires de données d'entraînement obtenus par échantillonnage avec remplacement (bootstrap) (Figure 3.5). Les prédictions de ces modèles sont ensuite agrégées, généralement par moyenne pour la régression ou par vote majoritaire pour la classification. L'exemple le plus célèbre de bagging est les forêts aléatoires (RF), où chaque arbre de décision est entraîné sur un échantillon bootstrap différent des données d'entraînement, et un sous-ensemble aléatoire de caractéristiques est considéré pour chaque division dans les arbres [271]. Cela réduit la variance et améliore la généralisation du modèle.

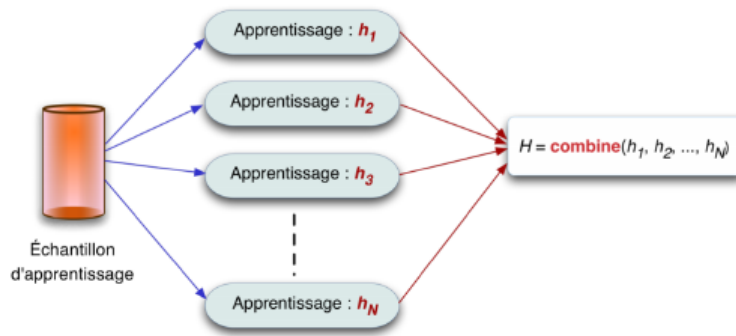


FIGURE 3.5 – Processus du Boosting de plusieurs modèles d'apprentissage.

3.4.5.5 Forêt aléatoire (RF)

L'algorithme des forêts aléatoires a été développé par Leo Breiman et Adele Cutler en 2001 [272]. Cet algorithme, dans sa version traditionnelle, réalise un apprentissage parallèle sur plusieurs arbres de décision générés aléatoirement et formés à partir de divers sous-ensembles de données. Le nombre optimal d'arbres peut aller jusqu'à plusieurs centaines.

La forêt aléatoire est constituée de chaque arbre sur un sous-ensemble aléatoire de données selon le principe du bagging, avec un sous-ensemble aléatoire de caractéristiques (voir Figure 3.6). Par la suite, les prédictions de ces divers arbres sont moyennées. L'algorithme des forêts aléatoires est considéré comme l'un des modèles les plus performants, avec peu de prétraitement des données. Les RFs se distinguent par l'ajout d'un élément aléatoire supplémentaire lors de la construction des arbres, où la caractéristique la plus avantageuse est sélectionnée parmi un ensemble aléatoire

de caractéristiques. Cela conduit à une grande variété dans le modèle, ce qui se traduit généralement par des performances améliorées.

Toutefois, la principale difficulté des RFs réside dans le fait que l'utilisation d'un grand nombre d'arbres peut entraîner une lenteur et une inefficacité excessive de l'algorithme pour les prédictions en temps réel. Malgré leur rapidité d'apprentissage, ces algorithmes peuvent être plutôt lents à fournir des prédictions une fois qu'ils ont été formés. Une plus grande précision demande un nombre plus élevé d'arbres, ce qui complique le modèle. L'algorithme des forêts aléatoires est assez rapide dans la plupart des applications pratiques, mais il peut y avoir des cas où des performances en temps réel sont essentielles et où d'autres méthodes seraient plus appropriées.

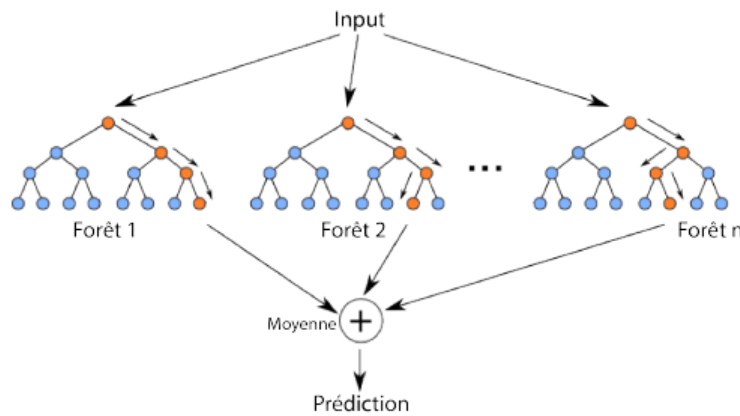


FIGURE 3.6 – Schéma de fonctionnement du modèle RF.

3.5 Solution logicielle proposée

3.5.1 Etude de l'existant

L'état de l'art des techniques d'approximation des valeurs de fitness dans les MOEAs est un domaine en constante évolution, offrant une gamme diversifiée d'approches pour résoudre les problèmes de multi-objectifs de manière efficace. Plusieurs travaux qui s'intéressent aux techniques d'approximation de fitness de substitution, pour approcher les valeurs de fitness et réduire le nombre d'évaluations directes données où le temps de calculs et la consommation de ressources mémoire sont très coûteux dans les problèmes compliqués. Au départ, ces techniques ont été développées

pour résoudre des problèmes mono-objectifs comme dans l'ouvrage cité dans [273]. Cependant, ces techniques étaient plus efficaces dans les MOPs, étant donné que la phase d'évaluation avec la fonction de fitness est la plus coûteuse en termes de calcul. Par conséquent, les techniques de substitution visent à remplacer ces fonctions de fitness par des modèles de calcul moins coûteux.

Dans [274], les auteurs ont utilisé une approche hybride dans laquelle un ensemble de réseaux ANNs et d'algorithmes KNN sont utilisés pour l'approximation des fonctions de fitness. Pour chaque génération après avoir classé les individus, un ensemble d'ANN est créé et formé (un ANN pour chaque classe). La limite de cette approche est l'apprentissage répétitif des ANN, qui consomme un temps de calcul considérable. De plus, dans certains cas, l'approximation de la valeur de fitness est erronée. Par conséquent, pour surmonter ce problème, Nain et Deb [275] proposent d'utiliser le contrôle évolutif pour améliorer les résultats des ANN, alternant entre l'évaluation directe et l'approximation de la valeur de fitness par les ANN. Cette méthode consiste à utiliser le GA avec une évaluation directe des individus pendant un nombre prédéfini de générations. La population résultant de cette exécution forme des données d'entraînement ANN qui seront utilisées pour approximer les valeurs de fitness des individus restants. L'inconvénient de cette méthode est de savoir comment définir la fréquence de contrôle de l'évolution. Dans [276], un modèle de processus gaussien a été appliqué comme mode de substitution pour accélérer l'EA.

Dans [277], les auteurs avaient appliqué un MOEA de tri non dominé élitiste avec un algorithme de régression à vecteur support comme modèle de substitution, cette hybridation leur a permis de remplacer la simulation de la consommation de temps dans l'évaluation du coût de ré-ordonnancement de la simulation. Sinon, dans [278], les auteurs ont utilisé une combinaison de deux méthodes, la première est l'algorithme de regroupement flou pour séparer la population en sous-groupes, et la seconde est le modèle d'approximation de la condition physique avec le MLP des ANN. Cependant, la mise en œuvre du modèle approché globalement sans combinaison avec la fonction de fitness d'origine pourrait générer une autre valeur optimale globale et ne pas converger vers la bonne valeur. Par conséquent, dans cette étude, ils ont proposé d'utiliser une combinaison des deux.

Dans [279] la méthode utilisée pour l'approximation était basée sur la distance

de Mahalanobis entre les individus, l'algorithme d'estimation de distribution (EDA) a été appliqué pour extraire des informations de base sur les individus hautement sélectionnés pour les utiliser plus tard dans la phase d'apprentissage de l'apprentissage algorithmique. L'application de la distance de Mahalanobis au lieu de la distance euclidienne dans l'algorithme KNN leur a permis de mesurer la distance relative en tenant compte des caractéristiques statistiques de la distribution des données. En utilisant l'algorithme distance-KNN de Mahalanobis, l'efficacité a été améliorée en mesurant la similarité entre les solutions qui ont plus d'informations structurelles.

Benatia et al [280] ont présenté une méthode basée sur la technique de regroupement et l'approximation de la forme physique des ANNs pour résoudre le problème du temps de calcul et du grand nombre de variables impliquées pour une optimisation du déploiement WSN dans les bâtiments intelligents de grande surface. Les ANNs appliqués ont été appris par les premières données de population et la méthode de regroupement a été appliquée pour regrouper la population en sous-population. Ils ont également introduit une fonction de fiabilité dans le calcul de la fonction objective de chaque individu pour réduire l'erreur générée par les ANNs. Le tableau 3.1 présente la répartition des travaux associés selon le type de variante des GAs, la technique d'approximation et le domaine d'application.

D'après les travaux cités, nous avons remarqué que les techniques de substitution sont fortement recommandées pour traiter des problèmes d'optimisation complexes avec des EAs. De plus, la majorité des études précédentes (tableau 3.1) étaient basées sur l'idée de construire un modèle global pour l'approximation des valeurs de fitness, et l'intégration d'algorithmes de regroupement pour améliorer leurs résultats. Mais malheureusement, l'utilisation de l'approximation de fitness peut dégrader la solution et introduire de faux optimums, surtout lorsque le nombre d'itérations de l'algorithme est très grand. De plus, l'application de modèles approximatifs de regroupement ignore la caractéristique originale de l'aspect fitness et sa signification physique.

Par conséquent, et après une grande étude dans ce sens, nous avons constaté que l'application des algorithmes de SML pour approximer la fonction de fitness à une fréquence d'entraînement fixe peut surmonter les lacunes précédentes. Étant donné que les variables de décision dans les COP sont de nature discrète, nous avons choisi

TABLE 3.1 – Répartition des travaux selon le type de variante des GAs, la technique d'approximation et le domaine d'application.

Travaux (année)	Type de variante GA	Outil d'approximation	Champs d'application
[274]	Algorithme génétique	ANN et KNN	Ackley, fonction Rosenbrock, fonction Sphère.
[275]	Algorithme génétique	ANN et contrôle évolutif	Problème d'ajustement de courbe B-spline.
[276]	Algorithme évolutif	Modèle de processus gaussien	Optimisation des profils de compresseurs de turbines à gaz stationnaires.
[277]	Tri élitiste non dominé MOEA	Algorithme de régression vectorielle (SVM)	Optimisation de la planification proactive face aux pannes stochastiques dans des environnements de production en détérioration.
[278]	Algorithme génétique	Perceptron multicouche et clustering flou	Fonction de benchmark pour optimisation globale.
[279]	Algorithme génétique	Distance KNN basée sur Mahalanobis	Session spéciale CEC 2013 sur l'optimisation des paramètres réels.
[280]	Algorithme génétique multi-objectifs	K-moyennes et ANN	Déploiement de nœuds WSN dans les bâtiments intelligents.

d'appliquer plusieurs algorithmes de régression pour approximer les valeurs de fitness, car par rapport aux autres techniques d'approximation, les algorithmes de SML ont donné de meilleurs résultats dans l'approximation des valeurs de fitness dans les COP, comme présenté dans [278]. De plus, le choix de ces algorithmes pour créer notre modèle de substitution était basé sur plusieurs raisons. Le plus important et à notre connaissance, ces outils de régression n'ont pas été appliqués auparavant pour le même problème d'optimisation dans le monde réel. Autre raison, néanmoins leurs qualités de stabilité et leurs capacités à manipuler des données volumineuses avec de nombreuses variables comme dans notre cas d'étude, ces algorithmes n'exigent pas un ajustement soigné des hyper-paramètres. Dans le cas du modèle approximatif RF, cet algorithme possède une compétence très puissante qui est l'évaluation de l'importance des caractéristiques. Cet outil de régression est très stable lorsqu'un nouveau point de données est introduit dans le jeu de données car un seul arbre est affecté, et comme le SPEA II est un algorithme stochastique, nous pouvons sûrement rencontrer des cas

similaires.

3.5.2 Description de la solution

Comme mentionné précédemment, l'hybridation propose un objectif de réduire le nombre d'évaluations directes dans le but de réduire le temps d'exécution, en les remplaçant par des estimations (approximatives) de la valeur de fitness des individus de la population. Les différentes phases du MOEA voir la génération des individus de la population initiale, mutation, croisement et la sélection restent les mêmes que dans le chapitre précédent, la phase d'évaluation est remplacée par une phase d'approximation des valeurs de fitness par les différents algorithmes utilisés qui sont entraîné en utilisant les données de la population initiale. Il y a aussi l'introduction d'une nouvelle phase, celle de l'apprentissage du modèle approximatif.

L'application des MOEAs pour l'optimisation du déploiement des nœuds dans un WSN destiné à la surveillance des sols est considérée comme un problème complexe, en partie en raison du temps de calcul nécessaire pour obtenir des solutions optimales. Les principaux défis associés à cette optimisation incluent :

- Grande Dimensionnalité : Le nombre de nœuds et la configuration possible des déploiements peuvent être très élevés, ce qui augmente de manière exponentielle la taille de l'espace de recherche.
- Contraintes Multiples : Il faut satisfaire plusieurs contraintes, comme la couverture complète de la zone surveillée, la connectivité du réseau, et la minimisation de la consommation d'énergie. Chaque contrainte ajoute de la complexité au problème.
- Buts Conflitants : Les objectifs d'optimisation sont souvent conflictuels. Par exemple, maximiser la couverture et minimiser la consommation d'énergie peuvent être des objectifs contradictoires, nécessitant des compromis délicats.
- Temps de Calcul Élevé : Évaluer la qualité des solutions potentielles peut nécessiter des simulations ou des calculs complexes, augmentant ainsi le temps de calcul global. Les MOEAs, en cherchant à explorer de multiples solutions simultanément, peuvent exacerber cette problématique en augmentant la charge computationnelle.

De ce fait, ce problème est un problème de conception complexe nécessitant une approche sophistiquée. Pour remédier à ces défis, nous avons opté pour une architecture d'hybridation de bas niveau [281]. Cette approche implique l'intégration d'une méthode d'approximation des valeurs de fitness directement dans le processus d'optimisation pour améliorer l'efficacité computationnelle et la performance globale de l'algorithme.

Nous avons choisi d'appliquer différents algorithmes de régression pour approximer les valeurs de fitness durant la phase d'évaluation dans SPEA II. Cette phase est particulièrement gourmande en temps de calcul, surtout lorsqu'il s'agit de grands problèmes avec plusieurs contraintes à satisfaire. L'idée est d'utiliser un modèle de substitution pour prédire les valeurs de fitness, ce qui permet de réduire le nombre d'évaluations directes coûteuses.

De ce fait, nous avons intégré de nouvelles étapes dans le déroulement du SPEA II. Les différentes étapes de l'approche hybride pour l'approximation des valeurs de fitness peuvent être présentées comme suit :

- **Évaluation Initiale de la Population** : La population initiale est évaluée en utilisant directement SPEA II. Chaque individu est évalué selon les critères définis, et les valeurs de fitness sont calculées.
- **Entraînement du Modèle d'approximation** : Les données obtenues de cette évaluation initiale, c'est-à-dire les individus et leurs valeurs de fitness correspondantes, sont utilisées pour entraîner un modèle de forêts aléatoires. Ce modèle sert de substitut pour prédire les valeurs de fitness pour de nouveaux individus dans les générations suivantes.
- **Utilisation du Modèle de Substitution** : Pour les générations suivantes, le modèle d'approximation prédit les valeurs de fitness des individus, ce qui permet d'éviter des évaluations directes coûteuses. Cela accélère considérablement le processus d'optimisation.
- **Ré-entraînement du Modèle d'approximation** : À des intervalles réguliers (contrôle évolutif), la population est évaluée directement par SPEA II pour mettre à jour les données de formation du modèle. Le modèle est ensuite ré-entraîné avec ces nouvelles données pour maintenir la précision des prédictions

et éviter les pièges des optima locaux. Cela assure que le modèle d'approximation reste aligné avec la réalité du problème à résoudre.

3.5.3 Définition du taux de contrôle de l'évolution

Le contrôle de l'évolution dans le contexte des algorithmes de régression fait référence à des mécanismes permettant d'assurer que le modèle reste précis et efficace tout au long de son utilisation, surtout lorsqu'il est utilisé dans des algorithmes évolutifs comme SPEA II. Le taux de contrôle du hybride nous permet de gérer la fréquence à laquelle notre modèle est ré-entraîner pour approximer les générations futures. En effet, pour visualiser l'impact qu'il a sur les qualités des solutions et le temps d'exécution des calculs, nous avons étudié son impact sur les résultats du déploiement des nœuds d'un WSN d'une parcelle agricole de 400 m² qui sont résumés dans le tableau 3.2.

TABLE 3.2 – Impact du taux de contrôle sur la qualité de la meilleure solution et le temps d'exécution.

Taux de contrôle	Nombre de cap- teurs (%)	Couv (%)	Sur- couv(%)	Con (%)	Dist (m)	Temps d'exécution (min)
2	32,44	99,93	77,84	100	169191,93	28,42
4	35,75	100,0	83,5	100,0	215570,07	16,03
8	36,46	99,56	82,57	100,0	216702,83	12,07
10	38,0	99,5	85,0	100,0,	245013,47	8,35
15	41,75	97,75	86,5	100,0	282785,09	5,93
20	38.5	99.25	88.0	100.0,	241583.05	5.51

On observe que pour des cadences de contrôle plus élevées, le temps d'exécution diminue considérablement, du fait de la diminution du temps de la phase d'évaluation. Cependant, cette diminution s'accompagne généralement d'une diminution de la qualité de la solution en raison des erreurs d'approximation produites par notre modèle comme illustré sur la Figure 3.7. De plus, comme mentionné dans le tableau 3.2 qui représente l'impact du taux de contrôle sur la meilleure qualité de la solution et le temps d'exécution. La somme pondérée du cas des objectifs de fitness est calculée en attribuant le même coefficient de pondération aux 5 objectifs, ce qui peut être une méthode d'analyse décisionnelle multicritère. Ainsi, si l'on compare le taux de contrôle de 2 et 20, on constate que les qualités de la solution ont diminué pour presque les cinq objectifs pour une même taille de parcelle. Par exemple, le taux de numérotation des nœuds capteurs est passé de 32,44 à 38,5. Pour l'objectif de sur-dénombrement, le taux est passé de 77,84 à 88,0. Par conséquent, le choix du taux de contrôle dépend de notre préférence, si nous voulons trouver une solution de qualité au détriment d'un temps d'exécution court, ou vice versa.

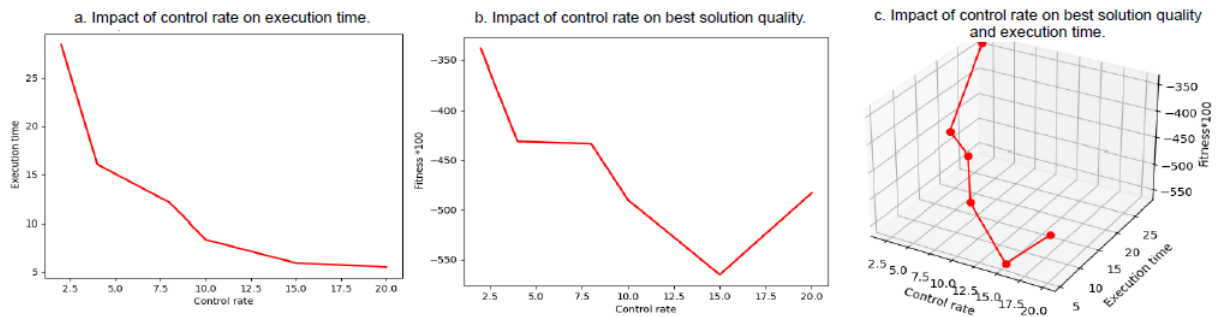


FIGURE 3.7 – Impact du taux de contrôle sur la meilleure qualité de la solution et le temps d'exécution.

3.5.4 Déroulement des Différentes Phases de SPEA II hybride avec Contrôle d'Évaluation

L'algorithme hybride SPEA II combine l'algorithme SPEA II avec différents algorithmes de régression pour l'optimisation multi-objectifs, notamment dans des contextes complexes tels que le déploiement de nœuds dans les WSNs. Voici une explication détaillée des phases de l'algorithme, incluant le contrôle d'évaluation.

3.5.4.1 Génération de la Population Initiale

Initialisation : Crée une population initiale d'individus, chacun représentant une solution potentielle au problème d'optimisation. Les individus sont générés de manière aléatoire ou selon une heuristique spécifique.

Évaluation Initiale : Chaque individu est évalué en utilisant les fonctions de fitness originales pour obtenir des valeurs de référence précises.

3.5.4.2 Formation Initiale des modèle de régression

Les données collectées lors de l'évaluation initiale sont utilisé pour la création des modèles. Ensuite, les données seront divisées en ensembles d'entraînement et de test pour évaluer la performance du modèle.

3.5.4.3 Phase Évolutive

Applique des opérations de mutation et de croisement pour générer de nouveaux individus à partir de la population existante, introduisant ainsi de la diversité génétique.

3.5.4.4 Évaluation avec Contrôle d'Évolution

Évaluation par les modèles d'approximation : Pour la majorité des générations, les modèle d'approximation prédit les valeurs de fitness des nouveaux individus. Cette évaluation est rapide et permet de gagner du temps de calcul. Évaluation par les Fonctions Originales : À intervalles réguliers définis par le taux de contrôle, les individus sont réévalués en utilisant les fonctions de fitness originales. Cela permet de maintenir la précision et d'éviter les optima locaux.

3.5.4.5 Ré-entraînement Continu du modèle d'approximation

Collecte de Nouvelles Données : Utilise les évaluations originales périodiques pour mettre à jour l'ensemble de données d'entraînement du modèle. Ré-entraînement : Entraîne de nouveau le modèle d'approximation avec les nouvelles données pour améliorer ses prédictions futures. Cela assure que le modèle reste pertinent et précis tout au long de l'exécution de l'algorithme.

3.5.4.6 Sélection

Archivage : Utilise la méthode SPEA II pour sélectionner les individus les plus adaptés pour la prochaine génération. Cela inclut l'utilisation d'un archive pour stocker les solutions non dominées trouvées jusqu'à présent. Sélection Basée sur la Fitness : Les individus sont sélectionnés en fonction de leurs valeurs de fitness (qu'elles soient prédites par le modèle d'approximation ou calculées par les fonctions originales) pour former la prochaine génération.

3.5.4.7 Boucle Itérative

Répétition des Phases : Les phases de mutation, croisement, évaluation, et sélection se répètent jusqu'à ce qu'un critère d'arrêt soit atteint (nombre maximum de générations, convergence des solutions, etc.). La Figure 3.8 montre le processus d'alternance entre l'utilisation du modèle de substitution et les évaluations directes par SPEA II. Les périodes de ré-entraînement sont clairement marquées pour illustrer le contrôle évolutif.

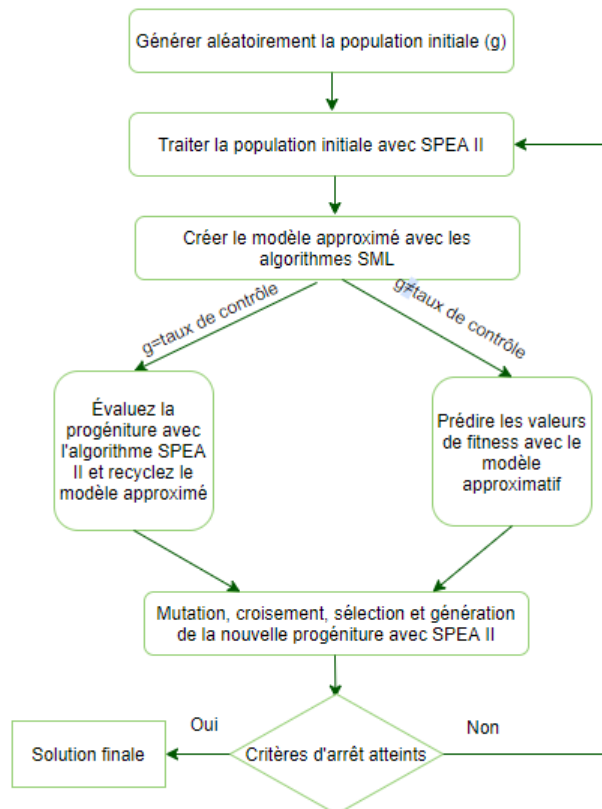


FIGURE 3.8 – Architecture de la solution logicielle.

Le pseudo-code (figure 3.9) décrit précisément les étapes de l'approche hybride combinant un modèle de substitution et l'algorithme évolutionnaire multi-objectifs RF-SPEA II. Il détaille comment chaque génération est évaluée, la manière dont le modèle de substitution RF est intégré pour estimer les valeurs d'objectifs et réduire le coût des évaluations réelles, ainsi que les conditions de déclenchement du ré-entraînement pour maintenir la précision.

Algorithm 1 RF-SPEA II Pseudo-code

```

Start
   $t \leftarrow 0$  ▷ Time initialization.
   $g \leftarrow N$  ▷ Le nombre d'initialisations de génération.
   $controlrate \leftarrow b$  ▷ Le modèle de substitution s'entraîne avec la fréquence b.
  Initialiser la population chromosomique  $P(g=0)$ 
  Évaluer  $P(g=0)$  en calculant ses cinq mesures de fitness,
  while  $g \leq N$  do
     $gn \leftarrow g + 1$  ▷ La génération suivante générée en appliquant des opérateurs génétiques.
    Sélectionner  $P(gn)$  à partir de  $P(g)$ 
    Croiser  $P(gn)$ 
    Mute  $P(gn)$ 
    if  $g == 1$  ou  $g \% == b$  then ▷ Évaluation directe des individus, création et formation du modèle de
      substitution à fréquence b.
      Évaluer  $P(gn)$ 
      Modèle = entraîner le modèle d'approximation avec  $P(gn)$  et leur évaluation
      Sfin = sélectionner les caractéristiques importantes de (modèle)
       $P(gn)\text{-trans} = \text{Transformer } P(gn) \text{ avec Sfin}$  ▷ Élimination des caractéristiques sans importance.
      Modèle = entraîner le modèle d'approximation avec  $P(gn)\text{-trans}$  et leur évaluation
    else ▷ Évaluation individuelle avec modèle de substitution.
      Prédire l'évaluation avec le modèle ( $P(gn)$ )
    end if
  end while
  Sortir les résultats vers une archive externe
End

```

FIGURE 3.9 – Pseudo-code du RF-SPEA II hybride.

Le déroulement des différentes phases de l'algorithme hybride, avec le contrôle d'évaluation, permet de maintenir un équilibre entre précision et efficacité. En utilisant des évaluations périodiques par les fonctions de fitness originales et en ré-entraînant continuellement le modèle d'approximation, l'algorithme assure une optimisation robuste pour des problèmes complexes comme le déploiement de nœuds dans les WSNs.

3.6 Implémentation de la solution logicielle

3.6.1 Cas d'étude

Pour évaluer le fonctionnement de notre modèle, nous avons choisi d'optimiser le déploiement des nœuds d'un WSN sur une parcelle agricole de 400 m^2 . Pour le modèle

d'approximation, nous avons utilisé quatre algorithmes de régression non linéaires, afin de mieux capturer la complexité de notre problème. Les algorithmes sélectionnés sont : les réseaux de neurones, les arbres de décision, les forêts aléatoires et les k-nearest neighbors. Nous avons appliqué ces algorithmes de régression sur notre ensemble de données et observé qu'ils produisaient des résultats variés en termes de précision et de performance. Dans la sous-section suivante, nous allons comparer ces résultats de manière détaillée, en analysant les forces et les faiblesses de chaque approche. Nous évaluerons la capacité de chaque algorithme à approximer les valeurs de fitness et à s'adapter aux contraintes spécifiques de notre problème, tout en tenant compte du temps de calcul et de l'efficacité générale de chaque méthode. Nous avons comparé les résultats obtenus avec l'algorithme SPEA II non modifié et l'algorithme hybride en utilisant les différents algorithmes de régression.

3.6.1.1 Définition des Hyper-paramètres

L'évaluation de notre modèle sur une parcelle agricole de 400 m^2 a nécessité une définition soigneuse des hyper-paramètres. Les tests ont montré que pour atteindre un optimum global tout en minimisant le temps d'exécution, une taille de population de 70 individus et un nombre de générations de 100 étaient adéquats. Cette configuration permet au modèle d'explorer efficacement l'espace de recherche et de converger vers une solution optimale.

Le tableau 3.3 présente une comparaison détaillée entre l'algorithme SPEA II non modifié et la version hybride du SPEA II avec les différents algorithmes de régression, en utilisant les mêmes hyper-paramètres. Cette comparaison porte sur deux aspects principaux : la qualité du plan de déploiement des nœuds d'un WSN et le temps d'exécution nécessaire pour atteindre ces plans. Les critères de qualité du déploiement incluent des mesures telles que la couverture réseau, la redondance des capteurs, la consommation d'énergie, et la robustesse globale du réseau par la somme pondérées des valeurs de fitness. En parallèle, le temps d'exécution est évalué pour chaque hybridation, afin de déterminer l'efficacité de chaque algorithme. Les résultats montrent comment chaque algorithme performe, offrant ainsi une vue d'ensemble de leurs avantages et inconvénients en termes de précision et de rapidité.

TABLE 3.3 – Comparaison entre SPEA II et SPEA II hybride avec les différents algorithmes de régression pour une taille de parcelle de 400 m^2 .

MOGAs	Temps d'exécution (s)	Nombre des capteurs (%)	couverture (%)	Sur-couverture (%)	Connectivité (%)	Distance (m)
SPEA II	1689.15	31.5	100.0	77.0	100.0	158833.64
MLP-SPEA II	308.52	35.36	72.37	68.88	72.74	14492.96
KNN-SPEAII	235.37	37.75	99.25	85.25	100.0	228178.84
DT-SPEA II	267.33	41.75	100.0	88.5	100.0	295071.56
RF-SPEAII	231.89	35.75	96.25	80.75	99.75	201080.29

3.6.2 Interprétation des résultats

L'hybridation de SPEA-II avec des algorithmes de régression peut améliorer les performances sur plusieurs critères. Voici une analyse détaillée des résultats obtenus par l'hybridation de SPEA-II avec divers algorithmes de régression.

Temps d'Exécution Le temps d'exécution est un critère crucial dans l'optimisation des WSNs, car il détermine l'efficacité temporelle de l'algorithme. SPEA-II pur présente le temps d'exécution le plus long à 1689.15 secondes, ce qui peut être prohibitif dans des scénarios réels nécessitant une réponse rapide. L'hybridation avec des algorithmes de régression réduit considérablement ce temps. Le modèle RF-SPEA II est le plus performant avec un temps de 231.89 secondes, suivi de près par KNN-SPEA II à 235.37 secondes et DT-SPEA II à 267.33 secondes. MLP-SPEA II, bien que plus lent que les autres modèles hybrides, réduit également le temps d'exécution à 308.52 secondes, ce qui représente une amélioration significative par rapport à SPEA-II pur. Ces résultats démontrent que l'hybridation améliore nettement l'efficacité temporelle de SPEA-II, rendant l'algorithme plus adapté aux applications pratiques.

- i. **Nombre de Capteurs** : Le nombre de capteurs déployés est un indicateur clé de l'efficacité et du coût d'un WSN. SPEA-II pur nécessite en moyenne 31.5

capteurs. L'hybridation avec des algorithmes de régression modifie ce nombre : DT-SPEA II en nécessite le plus avec 41.75 capteurs, ce qui peut augmenter les coûts et la complexité du déploiement. KNN-SPEA II et RF-SPEA II en nécessitent respectivement 37.75 et 35.75, ce qui est supérieur au SPEA-II pur mais peut être justifié par d'autres améliorations des performances. MLP-SPEA II utilise légèrement plus de capteurs que SPEA-II pur avec 35.36 capteurs. Le compromis entre le nombre de capteurs et les autres métriques de performance doit être soigneusement évalué en fonction des contraintes spécifiques de chaque application.

- ii. **Couverture** La **couverture** : est essentielle pour assurer que toutes les zones d'intérêt sont surveillées par les capteurs. SPEA-II pur, DT-SPEA II et KNN-SPEA II (à 99.25%) atteignent presque une couverture parfaite. RF-SPEA II suit de près avec une couverture de 96.25%. Cependant, MLP-SPEA II montre une couverture plus faible à 72.37%, ce qui peut être un inconvénient majeur si une couverture maximale est requise. Une couverture inférieure signifie que certaines zones peuvent ne pas être surveillées adéquatement, ce qui peut être critique dans certaines applications de sécurité ou de surveillance environnementale.
- iii. **Sur-couverture** : La sur-couverture indique une redondance excessive dans le déploiement des capteurs, ce qui peut entraîner une utilisation inefficace des ressources. MLP-SPEA II présente la sur-couverture la plus faible à 68.88%, ce qui est favorable. SPEA-II pur a une sur-couverture de 77%, tandis que RF-SPEA II, KNN-SPEA II et DT-SPEA II présentent des sur-couvertures de 80.75%, 85.25% et 88.5% respectivement. Une sur-couverture élevée peut signifier que plusieurs capteurs couvrent la même zone, ce qui peut être inutile et coûteux. L'équilibre entre une couverture adéquate et une sur-couverture minimale est crucial pour une optimisation efficace
- iv. **Connectivité** : La connectivité est vitale pour garantir que les données des capteurs peuvent être transmises efficacement à travers le réseau. SPEA-II pur, KNN-SPEA II et DT-SPEA II maintiennent une connectivité parfaite à 100%. RF-SPEA II suit de près avec 99.75%. Cependant, MLP-SPEA II présente une

connectivité réduite à 72.74%, ce qui pourrait poser des problèmes pour certaines applications nécessitant une connectivité élevée. Une connectivité réduite peut entraîner des pertes de données ou des retards dans la transmission des informations, compromettant ainsi la fiabilité du réseau.

- v. **Distance Totale** : La distance totale parcourue par les nœuds affecte directement la consommation d'énergie dans un WSN. MLP-SPEA II montre une amélioration spectaculaire avec une distance totale de 14492.96, contre 158833.64 pour SPEA-II pur. Cette réduction de la distance totale est bénéfique pour l'efficacité énergétique du réseau, prolongeant ainsi la durée de vie des capteurs. KNN-SPEA II, DT-SPEA II et RF-SPEA II présentent des distances totales plus élevées, avec DT-SPEA II ayant la plus grande distance totale à 295071.56. Une distance totale plus élevée peut indiquer que les capteurs sont déployés de manière moins optimale, augmentant la consommation d'énergie et réduisant la durée de vie du réseau.

L'hybridation de SPEA-II avec des algorithmes de régression montre des améliorations significatives en termes de temps d'exécution (Figure 3.10), avec RF-SPEA II étant le plus efficace. En termes de couverture et de connectivité, KNN-SPEA II et DT-SPEA II maintiennent des performances proches de SPEA-II pur. MLP-SPEA II se distingue par une réduction dramatique de la distance totale, ce qui est bénéfique pour l'efficacité énergétique, mais au détriment de la couverture et de la connectivité. Le choix de l'algorithme hybride dépendra des priorités spécifiques du déploiement WSN, qu'il s'agisse de minimiser le temps d'exécution, de maximiser la couverture et la connectivité, ou de réduire la distance totale pour une meilleure efficacité énergétique. En résumé, chaque méthode d'hybridation offre des avantages et des inconvénients spécifiques, et le choix optimal dépendra des exigences particulières de l'application et des contraintes opérationnelles du réseau de capteurs.

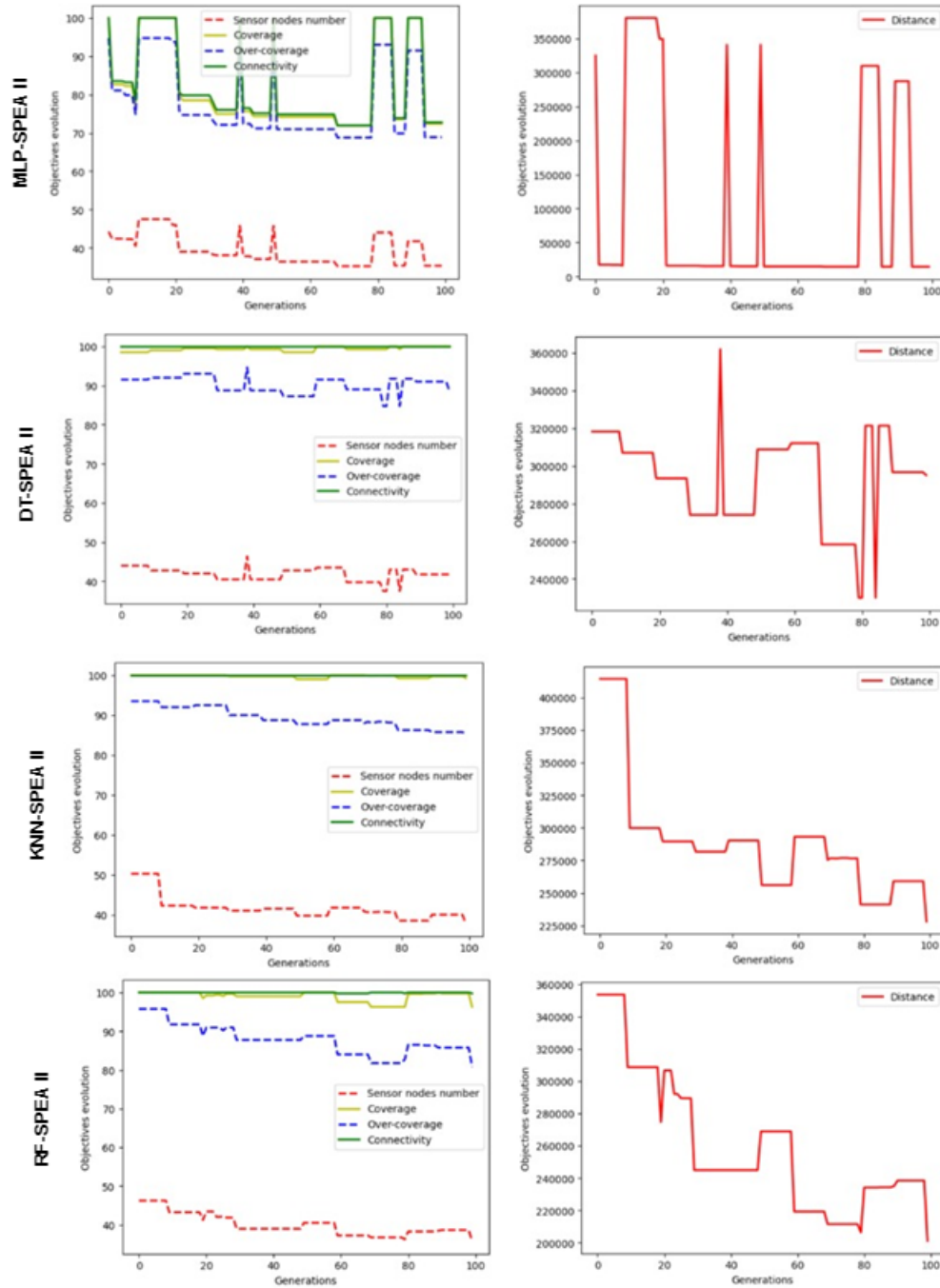


FIGURE 3.10 – Comparaison entre l'évolution des fonction objectives de SPEA II et SPEA II hybride avec les différents algorithmes de régression pour une taille de parcelle de 400 m^2 .

3.6.3 Comparaison entre les différents algorithmes de régression

L'utilisation des métriques d'évaluation est cruciale pour mesurer la performance des algorithmes de régression. De ce fait, nous avons utilisé la métrique de R-squared et la métrique de l'erreur moyenne quadratique.

3.6.3.1 Comparaison en utilisant la métrique de R-squared

L'image suivante présente l'évolution de la valeur de R-carré (R-squared) sur plusieurs générations pour différents modèles de régression utilisés dans l'hybridation de l'algorithme SPEA-II. En commençant par le MLP, on trouve que la valeur de R-squared commence très basse (en dessous de -40) et s'améliore rapidement pour atteindre 0 dès la deuxième génération. Par la suite, elle reste stable et très proche de 0. Cela indique que le modèle MLP atteint rapidement un point où il ne s'améliore plus significativement au fil des générations. Pour le DT algorithme, la valeur de R-squared fluctue entre des valeurs positives et négatives, avec des pics autour de 0. Les fluctuations montrent que le modèle de DT a des performances variables au fil des générations, avec des améliorations et des détériorations notables. De son côté, pour l'algorithme KNN, la valeur de R-squared commence autour de 0, puis fluctue également entre des valeurs positives et négatives, avec un creux significatif vers la génération 60. Ce comportement indique que le modèle KNN n'est pas stable et que ses performances varient fortement au fil des générations. Et à la fin, la valeur de R-squared de l'algorithme RF commence à 1, descend brusquement à un minimum autour de la génération 4, puis remonte progressivement vers une valeur positive proche de 1. Cette courbe suggère que le modèle Random Forest subit une dégradation significative de ses performances à un moment donné, mais finit par se stabiliser et s'améliorer.

En comparant les différentes courbes on trouve que concernant la stabilité des modèles, le MLP montre la meilleure stabilité après une amélioration initiale rapide. Cela suggère que MLP atteint un plateau de performance où il n'y a pas de fluctuations significatives par la suite. DT, KNN et RF montrent tous des fluctuations importantes, indiquant des variations dans leurs performances au fil des générations.

De plus, pour la performance moyenne des modèles, le RF commence et finit avec des valeurs positives de R-squared, suggérant une bonne performance moyenne malgré une dégradation intermédiaire. Le temps où le MLP finit près de 0, ce qui pourrait indiquer une performance moyenne plus faible comparée à RF. Le DT et le KNN présentent des performances moyennes plus variables et souvent négatives, ce qui peut suggérer une performance globale moins fiable. Et à la fin, par rapport à la résilience aux perturbations, le RF montre une certaine résilience en se redressant après une dégradation initiale. Le DT et KNN semblent moins résilients avec des fluctuations importantes et des périodes de performance réduite. Le MLP ne montre pas de perturbations significatives après l'amélioration initiale.

Par rapport à la métrique R-squared comme le montre la Figure 3.11, l'algorithme MLP présente une amélioration rapide et une grande stabilité, tandis que RF montre une bonne capacité à se redresser après des perturbations, offrant des performances finales positives. En revanche, DT et KNN montrent des performances très variables, indiquant une moins bonne fiabilité pour une optimisation stable. L'hybridation de SPEA-II avec RF ou MLP pourrait être préférée pour des performances plus constantes et prévisibles dans l'optimisation du déploiement des nœuds dans un WSN.

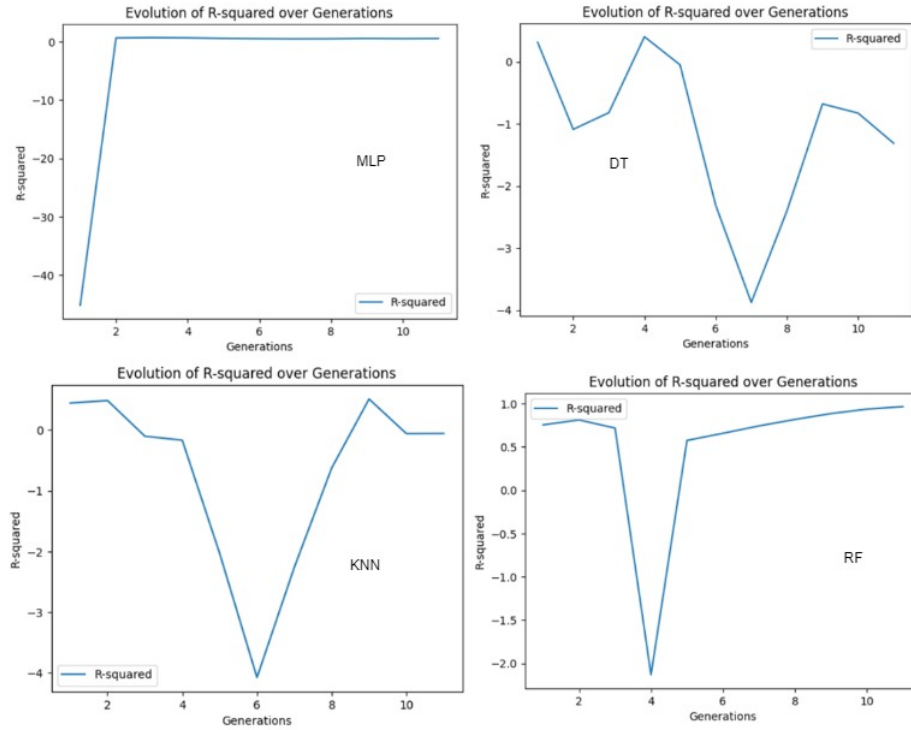


FIGURE 3.11 – Comparaison de la métrique de R-squared des différents algorithmes de régression pour une taille de parcelle de 400 m^2 .

3.6.3.2 Comparaison en utilisant la métrique de l'erreur moyenne quadratique

Les RMSE (Root Mean Squared Error) [282] sont calculés en prenant la racine carrée de la moyenne des carrés des erreurs entre les valeurs prédictives et les valeurs réelles. Un RMSE plus bas indique une meilleure performance prédictive. Cette mesure est cruciale pour évaluer la précision des modèles, avec des valeurs plus faibles indiquant de meilleures performances.

Nous analysons donc l'évolution du RMSE pour les quatre algorithmes au cours des générations (Figure 3.12). Commençant par le MLP qui montre une amélioration constante et significative, avec une réduction régulière du RMSE au fil des générations. La diminution progressive du RMSE indique que le modèle apprend efficacement des données, ajustant ses paramètres de manière à minimiser les erreurs de prédiction. La courbe descendante régulière suggère une convergence stable vers une solution optimisée. De plus, le KNN montre une amélioration impressionnante, réduisant le RMSE à une valeur proche de zéro. Cette diminution régulière et rapide indique que KNN

s'adapte extrêmement bien aux données, améliorant considérablement ses prédictions à chaque génération. La convergence vers une erreur proche de zéro souligne l'efficacité exceptionnelle de KNN dans ce contexte. Par rapport au DT, cet algorithme montre des fluctuations importantes tout au long des générations, avec des périodes de performance améliorée suivies de détériorations. Bien que le RMSE initial soit relativement bas, les pics indiquent des phases où le modèle perd en précision, notamment un pic significatif à la 6^{ème} génération. La stabilisation finale proche de la valeur initiale indique que le DT n'a pas réussi à converger vers une solution optimisée comme les autres algorithmes, suggérant une instabilité dans l'apprentissage. A la fin, dans ce qui concerne le RF il commence avec un RMSE relativement bas et montre des améliorations significatives, malgré quelques fluctuations intermédiaires. Le RMSE descend à une valeur inférieure à la 3^{ème} génération, ce qui indique une phase d'apprentissage efficace. Cependant, un pic à la 4^{ème} génération suggère une période de moins bonne performance avant de se corriger. La tendance générale vers une réduction du RMSE, se rapprochant de zéro, montre une amélioration continue et une capacité à surmonter les fluctuations pour converger vers une solution précise.

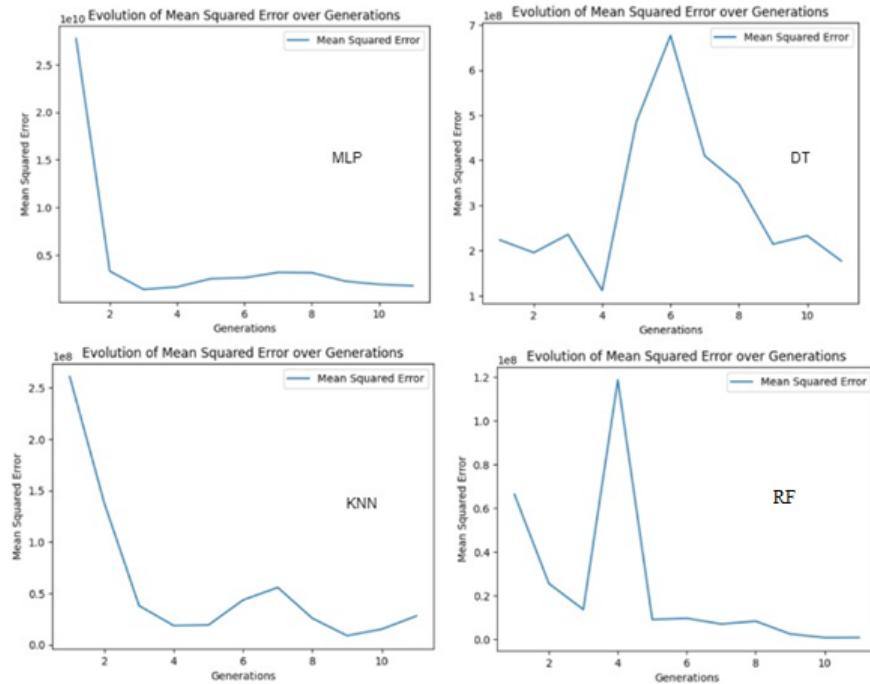


FIGURE 3.12 – Comparaison de la métrique de l'erreur moyenne quadratique des différents algorithmes de régression pour une taille de parcelle de 400 m^2 .

En comparant la performance de réduction du RMSE, le KNN montre la meilleure performance avec une réduction du RMSE à presque zéro, indiquant une convergence exceptionnelle. L'algorithme RF bien que fluctuant, il montre une amélioration continue avec une tendance finale vers une valeur très faible de RMSE, se rapprochant également de zéro. MLP : Montre une amélioration rapide avec une réduction significative du RMSE. Et à la fin, l'algorithme DT affiche des fluctuations importantes et une stabilisation finale proche de la valeur initiale, indiquant une instabilité et une performance moins optimale par rapport aux autres algorithmes. Par rapport à la stabilité de l'apprentissage les algorithmes MLP et KNN montrent une stabilité notable avec une réduction régulière du RMSE, suggérant un apprentissage efficace et une convergence stable. De sa part le RF malgré quelques fluctuations, montre une tendance générale positive vers une amélioration continue. Cependant l'algorithme DT montre des fluctuations importantes et des périodes de moins bonne performance, indiquant une instabilité dans le processus d'apprentissage.

En conclusion, bien que tous les algorithmes montrent des améliorations au fil des générations, KNN et RF se distinguent par leur capacité à réduire significativement le RMSE et à fournir des prédictions précises, avec KNN montrant une capacité exceptionnelle à presque éliminer l'erreur. MLP suit de près avec une amélioration régulière, tandis que DT affiche des performances moins stables et moins optimisées.

3.6.4 Amélioration des résultats obtenues par RF-SPEA II

En utilisant des métriques de comparaison telles que le R-squared et la RMSE, nous avons observé que l'algorithme RF démontre une capacité notable à se redresser après des perturbations [263], offrant ainsi des performances globalement positives. Bien que l'algorithme RF présente des fluctuations, il montre une tendance générale vers une amélioration continue, avec une diminution progressive du RMSE vers des valeurs très faibles, se rapprochant de zéro.

Cette tendance encourageante indique que l'algorithme RF est capable d'affiner ses prédictions de manière significative au fil du temps, malgré les variations initiales. En raison de cette performance prometteuse, nous avons décidé de concentrer nos efforts sur l'amélioration des performances de l'algorithme RF-SPEA II en explorant

deux axes principaux : la sélection des caractéristiques et l'ajustement des hyperparamètres.

3.6.4.1 Sélection de caractéristiques

La sélection des caractéristiques est une étape cruciale dans la conception et l'optimisation des algorithmes d'apprentissage automatique, en particulier dans le contexte de l'approximation des valeurs de fitness. Une sélection adéquate des caractéristiques permet d'améliorer la performance du modèle et d'éviter le sur-ajustement.

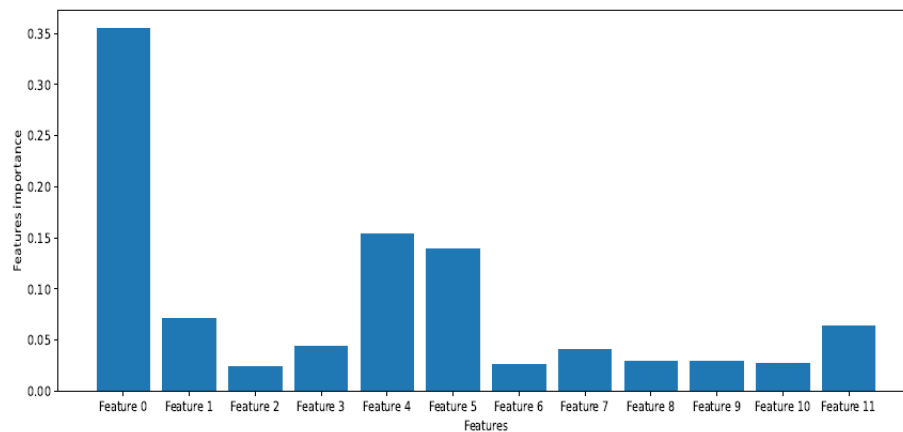
Tout d'abord, un ensemble de forêts aléatoires est construit en utilisant un échantillon aléatoire de données et un sous-ensemble aléatoire de caractéristiques à chaque division de l'arbre. Par la suite, pendant la construction de chaque arbre, l'algorithme mesure l'importance de chaque caractéristique en calculant l'augmentation de l'impureté (GINI ou entropie) [272] causée par chaque division basée sur cette caractéristique. Plus cette augmentation est élevée, plus la caractéristique est considérée comme importante pour la classification ou la régression. Une fois que le modèle de forêts aléatoires est entraîné, l'importance de chaque caractéristique est calculée en moyennant les mesures d'importance de chaque arbre dans la forêt. Les caractéristiques avec une importance plus élevée sont considérées comme plus utiles pour la prédiction des valeurs de fitness. A la fin, un seuil d'importance peut être défini pour sélectionner les caractéristiques les plus pertinentes. Seules les caractéristiques dont l'importance dépasse ce seuil sont conservées pour l'analyse ultérieure.

Dans notre cas, nous avons choisi de calculer la diminution moyenne des impuretés (Figure 3.13) et de classer les caractéristiques par ordre d'importance. Cette approche nous a permis de ne conserver que les caractéristiques les plus importantes qui fournissent une importance cumulée d'au moins 95%, pour garantir que notre modèle reste performant. Cette métrique d'évaluation est également connue sous le nom l'impureté de GINI [272]. En utilisant un graphique comme celui-ci (Figure suivante), nous pouvons décider où placer un seuil, en faisant un compromis entre les performances de notre modèle et le nombre de fonctionnalités utilisées. Cette approche supprime les obstacles insignifiants qui n'affectent pas trop la sortie de notre modèle mais peuvent diminuer les performances en introduisant du bruit. Dans notre problème, un caractère

TABLE 3.4 – Comparaison entre RF-SPEA II et RF-SPEA II (GiNI) pour une taille de parcelle de 400 m^2 .

MOGAs	Temps d'exécution (s)	Nombre des capteurs(%)	couverture (%)	Sur-couverture (%)	Connectivité (%)	Distance (m)
RF-SPEAII	231.89	35.75	96.25	80.75	99.75	201080.29
RF-SPEAII (GINI)	276.50	32.5	94.5	77.0	100.0	172387.94

n'est rien d'autre qu'un gène qui compose un chromosome, c'est-à-dire un individu. Moins de fonctionnalités signifient généralement des modèles de formation plus rapides.

FIGURE 3.13 – Importance des caractéristiques avec le le modèle approximatif RF pour un exemple de taille de parcelle de 12 m^2 .

Le tableau 3.4 et la Figure 3.14 compare deux versions de l'algorithme RF-SPEA II en termes de performances. RF-SPEA II est plus rapide (231,89 secondes contre 276,50 secondes pour RF-SPEA II (GINI)) et offre une meilleure couverture (96,25 % contre 94,5 %), mais utilise un peu plus de capteurs (35,75 contre 32,5). RF-SPEA II (GINI) est légèrement supérieur en termes de connectivité (100 % contre 99,75 %), tandis que RF-SPEA II a une sur-couverture un peu plus élevée (80,75 % contre 77,0 %). comme le montre la figure suivante.

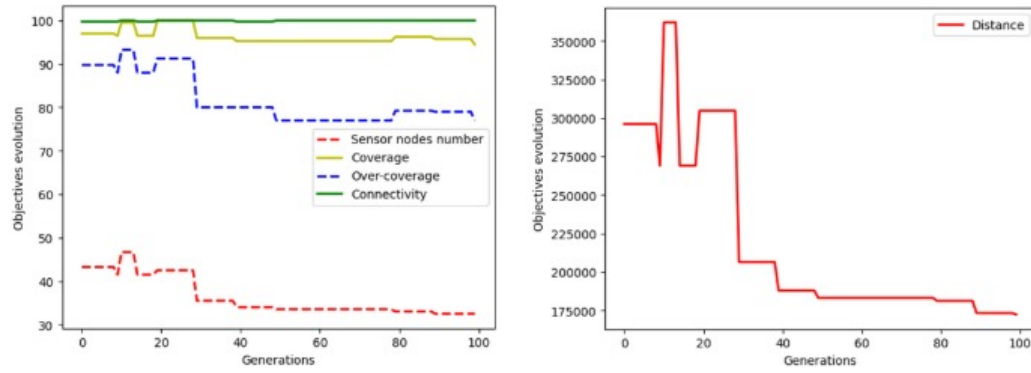


FIGURE 3.14 – L'évolution des fonction objectives du RF-SPEA II (GINI) pour une parcelle agricole de taille de 400 m^2 .

Ces graphiques suivants (Figure 3.15) montrent que, bien que l'algorithme RF-SPEA II utilisant l'importance de GINI puisse rencontrer des perturbations importantes au cours de son exécution, il a la capacité de se rétablir et d'améliorer ses performances globales. La chute drastique du R^2 autour de la 7ème génération pourrait être due à une défaillance temporaire dans l'adaptation du modèle aux données, mais l'amélioration subséquente des deux métriques (MSE et R^2) montre que l'algorithme réussit finalement à corriger sa trajectoire.

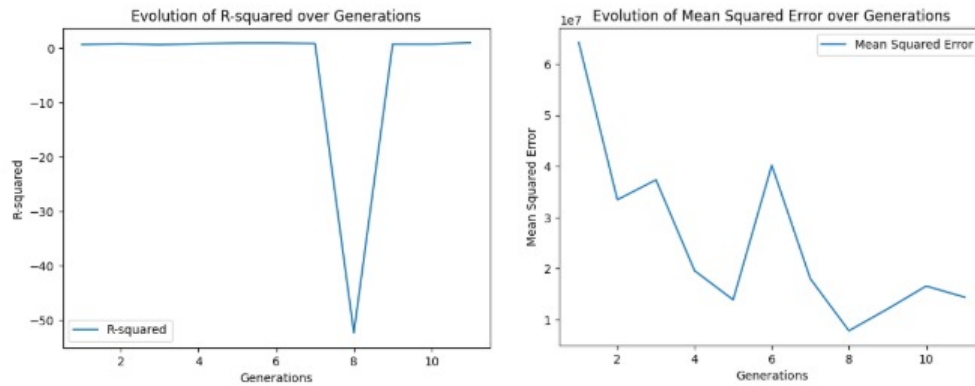


FIGURE 3.15 – Évolution des Performances de l'Algorithme RF-SPEA II (GINI) au Fil des Générations

3.6.4.2 Ajustement des Hyper-paramètres

Pour améliorer la qualité des solutions, la taille de la population et le nombre de générations ont été ajustés, offrant ainsi plus d'opportunités au modèle RF de s'entraîner correctement et d'explorer l'espace de recherche de manière plus exhaustive.

TABLE 3.5 – Comparaison entre RF-SPEA II (a) et RF-SPEA II (b) avec des Hyper-paramètres différents pour une taille de parcelle de $400 m^2$.

MOGAs	Temps d'exécution (s)	Nombre des cap-teurs(%)	couverture (%)	Sur-couverture (%)	Connectivité (%)	Distance (m)
RF-SPEAII (a)	231.89	35.75	96.25	80.75	99.75	201080.29
RF-SPEA II (b)	570.78	31.75	100.0	78.75	100.0	166633.34

Le tableau 3.5 présente une comparaison des performance de RF-SPEA II pour des différents hyper-paramètres et une taille de parcelle de $400 m^2$.

En augmentant le nombre de générations, les solutions obtenues étaient de meilleure qualité tout en maintenant un temps d'exécution meilleur que celui de l'algorithme SPEA II non modifié. comme le montre le tableau précédent, la distance de séparation des nœuds capteurs a diminué de 151859,30 à 148471,08, le taux de sur-couverture a diminué de 77,25 à 69,75, comme illustré dans la Figure 3.16.

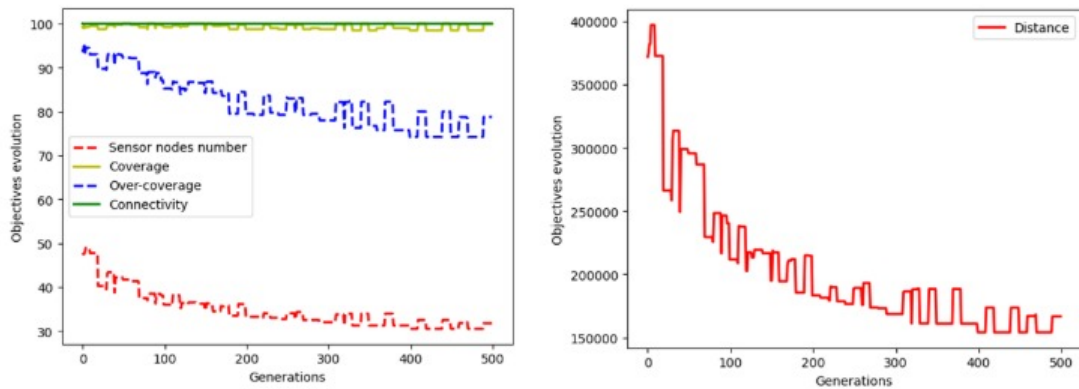


FIGURE 3.16 – L'évolution des fonctions objectives de RF-SPEA II pour différents hyper-paramètres.

Comparé à l'étude présentée dans [280], notre approche présente plusieurs améliorations notables. L'étude [280] a montré que leur modèle de déploiement pouvait trouver de bons résultats cinq fois plus rapidement que leurs autres méthodes de déploiement. Cependant, cette rapidité a été obtenue au détriment de la qualité des

solutions de déploiement, qui étaient inférieures. Un problème majeur était qu’après un certain nombre d’itérations, une erreur introduite par l’estimateur lors du calcul de la valeur de fitness entraînait de fausses approximations, compromettant ainsi les résultats finaux.

Dans notre étude, nous avons observé que le temps d’exécution de notre algorithme RF-SPEA II hybride était presque six fois plus rapide pour les mêmes hyperparamètres, comparé au SPEA II non modifié. Cette amélioration de la vitesse est significative, mais ce n’est pas le seul avantage. Nous avons intégré un mécanisme de contrôle d’évolution pour éviter les erreurs constatées dans l’étude [280]. Ce mécanisme implique un réentraînement périodique du modèle d’approximation à des intervalles de génération spécifiques. Cette stratégie permet de réduire le risque de convergence vers de faux optimums en assurant que le modèle d’approximation reste précis tout au long du processus d’optimisation. Pour valider l’efficacité de notre modèle, nous avons utilisé le R-squared et la RMSE comme métriques clés. Le RMSE compare la valeur de la fonction de fitness d’origine à celle estimée par le modèle approximatif, fournissant une mesure de la précision de notre approximation. Comme illustré dans la Figure 3.17, nous avons réussi à maintenir le RMSE à un niveau relativement bas pour les cinq objectifs à chaque entraînement de notre algorithme d’approximation. Cette précision nous a permis de mettre à jour chaque individu avec des valeurs de fitness approximatives qui ont conduit à une convergence vers des solutions optimales.

En conclusion, notre approche non seulement accélère considérablement le temps d’exécution, mais assure également que les solutions de déploiement restent de haute qualité, grâce à un mécanisme de contrôle d’évolution et une utilisation efficace des métriques pour valider et affiner notre modèle d’approximation.

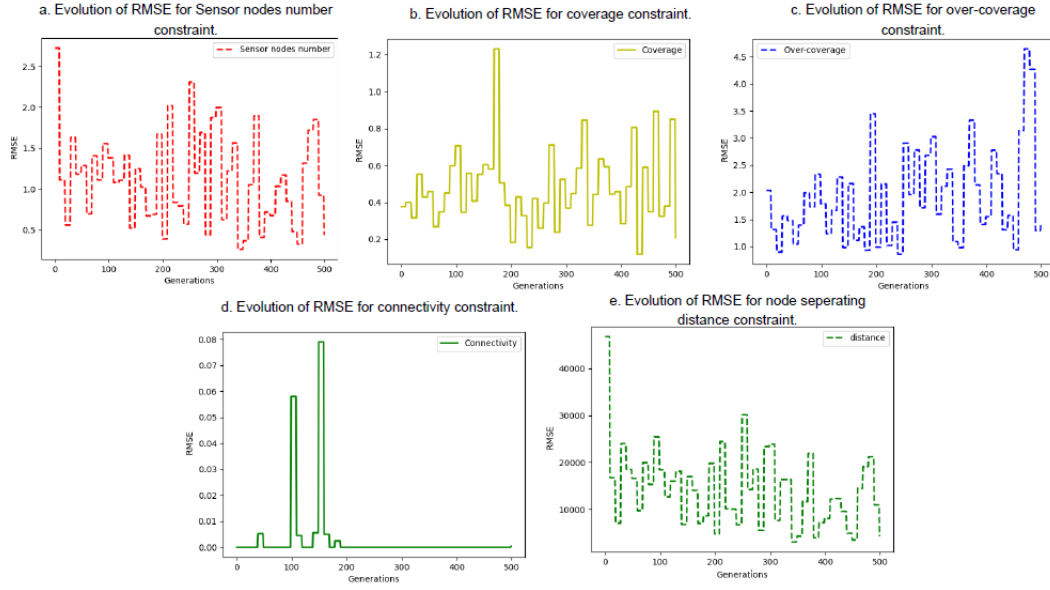


FIGURE 3.17 – Évolution du RMSE selon les générations pour une taille de parcelle de 400 m^2 .

Partie 2 :

Accélération matérielle des MOEAs : parallélisations du modèle proposé

3.7 Modèle de parallélisations des MOEAs

Les problèmes d'optimisation deviennent de plus en plus complexes et leurs besoins en ressources ne cessent d'augmenter. Les problèmes d'optimisation du monde réel sont souvent NP-difficiles pour la consommation de ressources CPU et/ou la mémoire. Bien que l'utilisation des EA réduise considérablement la complexité de calcul du processus de recherche. Ce dernier reste un utilisateur chronophage pour de nombreux problèmes dans divers domaines d'application, où la fonction objectif et les contraintes associées au problème sont gourmandes en ressources (ex. CPU, mémoire) et la taille de l'espace de recherche est très élevée [283]. Dans cette optique et pour atteindre deux objectifs principaux, le premier est de résoudre le problème de la consommation de temps en répartissant l'effort de calcul et le second est de bénéficier d'un parallélisme du point de vue algorithmique, en analogie avec l'évolution parallèle naturelle de populations réparties dans l'espace, plusieurs tentatives et modèles parallèles ont été proposés. L'objectif de cette section est de présenter les meilleurs modèles parallèles d'AE connus.

3.7.1 Origine et parallélisme des EAs

D'un point de vue algorithmique, la principale source de parallélisme pour les EA est l'exécution simultanée de leurs itérations de boucle interne. Malheureusement, c'est souvent la seule source facilement accessible dans les EA, car la plupart des autres étapes dépendent du temps et nécessitent le calcul des étapes précédentes pour être complété. Même lorsque le parallélisme est disponible, la synchronisation dépendante du temps des étapes EA donne des latences importantes, rendant le calcul parallèle non pertinent [284].

Une EA contient plusieurs étapes qui peuvent être indépendantes ou non. Pour commencer, l'initialisation de la population est intrinsèquement parallèle car tous les individus sont créés indépendamment (généralement avec des valeurs aléatoires). Ensuite, tous les individus nouvellement créés doivent être évalués. Mais, comme ils sont tous évalués indépendamment à l'aide de la fonction de fitness, l'évaluation peut également être effectuée en parallèle. Il est intéressant de noter que dans les évaluations

environnementales, l'évaluation des individus est généralement l'étape la plus consommatrice de l'évaluation environnementale.

Une fois la population parentale obtenue (par évaluation de tous les individus de la population initiale), il faut sélectionner un échantillon de parents sur lequel seront appliqués les opérateurs de variation. Dans les EA, la sélection des parents est également parallélisée car un parent peut être sélectionné plusieurs fois, ce qui signifie que des sélecteurs indépendants peuvent choisir ce qu'ils veulent, sans aucune restriction.

La création d'un descendant à partir des parents sélectionnés est également une étape complètement indépendante : un opérateur de croisement doit être appliqué aux parents, suivi d'un opérateur de mutation sur le descendant créé. Ainsi, on peut voir que jusque-là (maintenant), toutes les étapes de la boucle évolutive sont intrinsèquement parallèles, sauf la dernière, le remplacement. Afin de préserver la diversité dans les générations successives, la génération ($N + 1$) ème est créée en sélectionnant un des meilleurs individus de la population des enfants des parents + la population des enfants de la N ème génération.

Cependant, si un individu est autorisé à apparaître plusieurs fois dans la nouvelle génération, il pourrait rapidement devenir prééminent dans la population, induisant ainsi une perte de diversité qui réduira le pouvoir d'exploration de l'algorithme [285].

Par conséquent, les évaluations environnementales exigent que tous les individus de la génération soient différents. Il s'agit d'une réelle restriction au parallélisme, car cela signifie que la sélection des N survivants peut se faire indépendamment, sinon un même individu peut être sélectionné plusieurs fois par plusieurs sélecteurs indépendants.

D'autre part, une quantité importante de parallélisme peut résider dans le domaine du problème à résoudre ou dans l'espace de recherche correspondant. En effet, il n'y a pas de dépendances de données entre le coût ou les fonctions d'évaluation des différentes solutions, elles peuvent donc être évaluées en parallèle. De plus, théoriquement, le parallélisme dans la solution ou l'espace de recherche est aussi grand que l'espace lui-même. Il existe des limites considérables à l'exploitation efficace de ce parallélisme. Cependant, pour des raisons évidentes, affecter un processeur à chaque évaluation de la solution n'a aucun intérêt. L'espace de solutions ou de recherche doit être partagé entre les processeurs, par conséquent, une sérialisation de l'évaluation des solutions affectées au niveau d'un même processeur doit avoir lieu. Les partitions

résultantes sont généralement encore trop grandes pour une énumération explicite et, par conséquent, une méthode de recherche exacte ou heuristique est toujours nécessaire pour l'explorer implicitement.

Le partitionnement pose deux questions au regard de la stratégie globale de l'EA [284] : la maîtrise d'une recherche globale menée séparément sur plusieurs partitions de l'espace d'origine et la complétude de la solution finalement atteinte. L'allocation des ressources doit conduire à une exploration efficace, en évitant, par exemple, de trouver des régions avec des solutions de mauvaise qualité. Enfin, on pouvait se demander si plusieurs générations pouvaient évoluer en parallèle, le fait que la génération (N+1) soit basée sur la génération N invalidait l'idée.

3.7.2 Classification des modèles d'EA parallèles

Nous nous intéressons à présent à leur utilisation dans le contexte des optimisations du monde réel. Dans ce type d'applications, l'évaluation est souvent effectuée à travers de lourdes simulations très coûteuses. Ces algorithmes nécessitent donc d'être parallélisés afin de pouvoir réaliser les optimisations en un temps raisonnable. Plusieurs modèles de parallélisations des algorithmes évolutionnaires ont été proposés dans la littérature. Ces modèles qui diffèrent dans le cœur de la parallélisations, peuvent être classés en trois catégories (Figure 3.18) : les modèles "maîtres-esclaves", les modèles en îlots et les modèles totalement distribués [286, 287].

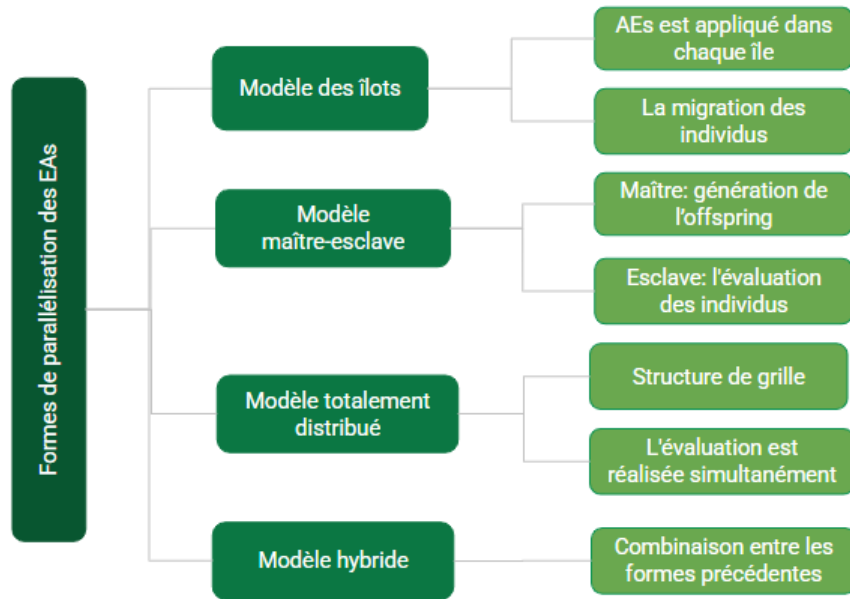


FIGURE 3.18 – Les formes de parallélisations des EAs.

3.7.2.1 Modèle Maître-esclave

Dans le modèle "maître-esclave", qui est le modèle le plus simple à mettre en œuvre, le processus maître distribue l'évaluation de la fonction objective sur les différents nœuds esclaves, et effectue toutes les étapes de l'algorithme évolutionnaire (sélection, croisement, mutation). La communication entre les individus n'a lieu qu'après que les différents esclaves retournent l'évaluation qui leur a été assignée. Il est de ce fait identique algorithmiquement parlant, à un algorithme évolutionnaire séquentiel. Le modèle maître esclave a été largement utilisé dans la littérature. Les premiers travaux remontent aux années 70 avec Bethke [288] qui a été le premier à décrire une implémentation parallèle d'un AE. Par la suite, Grefenstette [191] proposa plusieurs prototypes des AE parallèles représentant plusieurs variantes du modèle maître esclave. Sachant que le coût d'optimisation global avec ce modèle comporte deux temps, celui de l'évaluation des différents enfants et le temps de communication entre les différents nœuds esclaves et le processus maître, des travaux plus récents dans la littérature se sont intéressés à la problématique du coût de communication élevé qui peut nuire à l'efficacité de l'algorithme [289–291]. Dans ce modèle, l'accélération en temps de calcul est prouvée être linéaire relativement au nombre de processeurs, jusqu'à une certaine limite (quand les temps des communications deviennent plus

important que le temps d'une évaluation [286]. A noter que dans le cas des applications réelles où l'évaluation de la fonction objectif est très couteuse, les temps des communications sont souvent négligés.

3.7.2.2 Modèle en îlots

Ce modèle de parallélisations concerne les algorithmes évolutionnaires avec plusieurs populations. Il tient son appellation du parallèle fait avec les populations naturelles qui sont relativement isolées les unes des autres, formant des "îles". Toutefois, les individus peuvent "migrer" occasionnellement à une autre population. Les premiers travaux dans ce domaine remontent aux années 60 avec Bossert [292] qui fut probablement le premier à avoir proposé un algorithme évolutionnaire avec des populations multiples afin d'améliorer la qualité des solutions dans un problème d'optimisation. D'autres travaux qui ont été proposés par la suite [293, 294] diffèrent selon les choix faits sur les nombreux paramètres à piloter avec ce modèle comme par exemple : le nombre et la taille des îles, la fréquence de migration, le nombre et la destination des migrants, et enfin la méthode utilisée pour sélectionner quel individu va migrer. D'ailleurs c'est la raison pour laquelle ce type de parallélisations reste difficile à contrôler.

3.7.2.3 Modèle totalement distribué

Dans le modèle totalement distribué, la population est structurée en une grille rectangulaire bidimensionnelle où chaque nœud comprend un ou très peu d'individus, et l'évaluation de la fonction objective est réalisée simultanément pour tous les individus. Les opérations de sélection et de croisement sont réalisées avec les nœuds voisins. Ce type de parallélisations des algorithmes évolutionnaires peut être croisé dans la littérature sous d'autres appellations comme par exemple : EA de diffusion [295], AE cellulaires [Whitley 1993] [Tomassini 1999] ou encore EA massivement parallèles [296]. Cependant, ce dernier modèle reste peu étudié dans la littérature en comparant avec les deux premiers modèles.

3.7.3 Relation entre le modèle EA parallèle et l'architecture parallèle ciblée

Généralement, la question la plus importante pour tout modèle parallèle concerne la configuration de calcul sous-jacente que nous devrions utiliser ? Cette question n'a pas de réponse unique en raison de la difficulté de conception des modèles généraux de programmation parallèle en raison de l'espace architectural important qu'ils doivent couvrir et de la résistance représentée par les paradigmes et langages de programmation actuels. En tout état de cause, de nombreux problèmes importants sont suffisamment réguliers dans leurs dimensions spatiales et temporelles pour se prêter au calcul parallèle ou distribué et les algorithmes évolutionnaires en font certainement partie. Pour ces raisons, cette partie a été proposée et a pour objectif, une brève description des modèles d'architectures parallèles existants, puis la correspondance entre l'architecture sous-jacente et le modèle d'EA parallèle.

3.7.4 Architectures sous-jacentes existantes : un aperçu

Le parallélisme peut se produire à plusieurs niveaux dans les systèmes informatiques : niveau tâche, niveau instruction ou à un niveau machine inférieur. Bien qu'il existe plusieurs façons de classer les architectures parallèles, le modèle Flynn standard est largement accepté comme point de départ. Mais le lecteur doit être conscient qu'il s'agit d'une classification approximative : par exemple, même les processeurs série d'aujourd'hui sont en réalité très parallèles dans la manière dont ils exécutent les instructions, ainsi qu'en ce qui concerne l'interface mémoire. Même à un niveau architectural supérieur, de nombreuses architectures parallèles sont en fait des hybrides des classes de base. Pour un traitement complet du sujet. La taxonomie de Flynn [297] est basée sur le concept d'instructions et de flux de données. Il existe quatre combinaisons possibles appelées classiquement SISD (instruction unique, flux de données unique), SIMD (instruction unique, flux de données multiples), MISD (instruction multiple, flux de données unique) et MIMD (instruction multiple, flux de données multiples). Une autre décision de conception importante est de savoir si la mémoire système s'étend sur un seul espace d'adressage ou si elle est distribuée dans des segments séparés qui sont adaptés individuellement. Le premier type est appelé mémoire

partagée tandis que le second est appelé mémoire distribuée. Ce n'est qu'une subdivision logique indépendante de la façon dont la mémoire est physiquement construite. Les multi-ordinateurs à mémoire distribuée sont également une architecture populaire qui convient bien à la plupart des charges de travail parallèles. Étant donné que les espaces d'adressage de chaque processeur sont séparés, la communication entre les processeurs doit être mise en œuvre via une forme de transmission de messages. À cette classe appartiennent les ordinateurs en réseau, parfois appelés grappes d'ordinateurs. Un autre concept attrayant est le World Wide Web qui fournit des infrastructures importantes pour l'informatique distribuée. Comme elle implémente un modèle général de calcul distribué, cette technologie Web peut être utilisée pour le calcul parallèle et pour les applications informatiques et liées à l'information. L'exploitation du Web ou d'une autre ressource informatique distribuée géographiquement de manière à ce qu'elle ressemble à un seul ordinateur pour l'utilisateur s'appelle un méta-calculateur. L'environnement Java est une étape importante dans cette direction. Chacun des paradigmes décrits ci-dessus a ses forces et ses faiblesses. Le lecteur est invité à se référer à d'autres articles pour plus d'informations [298]. Dans la partie suivante, une présentation sur la façon dont les différentes architectures peuvent être utilisées pour les EA distribués aura lieu.

3.8 Calcule distribué

Un calcul parallèle est défini comme l'exécution d'un ou plusieurs programmes, simultanément, par plusieurs processeurs [299]. Nous avons, en général, deux manières de réaliser un calcul parallèle. La première consiste à découper le programme en plusieurs tâches de calcul puis, exécuter toutes ces tâches en parallèles par différents processeurs. La seconde nécessite le partitionnement des données du problème à traiter, de manière à ce que chaque partie de données soit attribuée à un processeur différent. Ensuite, tous les processeurs exécutent en parallèle les instructions du même programme mais en opérant sur des données différentes. Cette dernière méthode, appelée la parallélisation de données. Le calcul parallèle a pour objectif d'exploiter la grande quantité de ressources (processeurs, mémoires, espaces de stockage, etc) qu'offrent les

calculateurs parallèles ; ceci, dans le but de réduire le temps d'exécution des applications nécessitant un long traitement et/ou pour pouvoir exécuter celles portant sur des volumes de données très importants. Tout cela nous permet d'aborder de nouveaux problèmes, de plus en plus, complexes et de tailles toujours croissantes.

3.8.1 Eléments de parallélisme

3.8.1.1 Mémoires des architectes parallèles

Nous pouvons distinguer, en général, deux modèles de gestion de la mémoire des calculateurs parallèles : la mémoire partagée et la mémoire distribuée. Ces deux modèles de mémoire permettent de définir les modalités d'accès aux données des autres processeurs dans un calcul parallèle.

- **Mémoire partagée** : Dans ce type d'architecture, les processeurs du calculateur parallèle ont un accès direct au même espace mémoire physique via des liens de communication performants, avec un temps d'accès rapide et équitable [300] (Figure 3.19). En effet, les processeurs peuvent opérer Indépendamment mais toutes les données du calcul parallèle sont placées dans une mémoire commune et ce, de façon à ce que les changements établis dans la mémoire par un processeur soient immédiatement visibles par les autres processeurs. Dans ce cas, les échanges de données entre processeurs sont effectués via la synchronisation des lectures/ écritures dans la mémoire partagée.

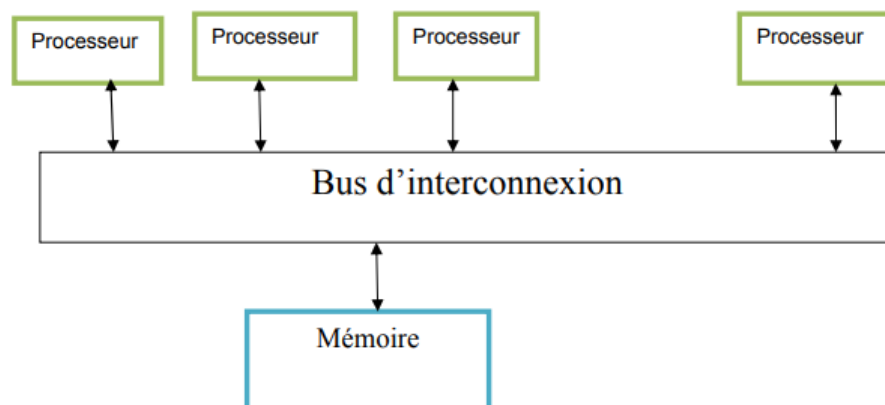


FIGURE 3.19 – Exemple d'architecture parallèle à mémoire partagée.

- **Mémoire distribuée** : Nous pouvons trouver ce type de mémoire, plus particulièrement, sur les plateformes de calcul parallèle à ressources distribuées, par exemple les grappes et les grilles de calcul. Dans ce cas, chaque processeur de la plateforme parallèle possède sa propre mémoire locale dans laquelle les changements ne sont pas visibles depuis les autres processeurs. Par conséquent, l'accès aux données des mémoires distantes (mémoires des processeurs voisins) est assuré par des envois de messages entre processeurs via un réseau de communication. La Figure 3.19 illustre un exemple d'architecture parallèle à mémoire distribuée[301].

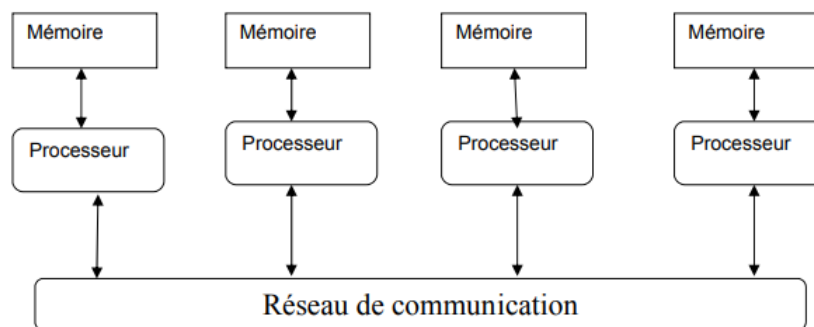


FIGURE 3.20 – Exemple d'architecture parallèle à mémoire distribuée.

3.8.1.2 Implémentations d'EA parallèles

Le processus d'évolution artificielle peut être implémenté sur du matériel parallèle de diverses manières, l'idée de base étant que les individus de la population sont indépendants les uns des autres et ne sont qu'à quelques étapes de l'algorithme qui nécessitent une interaction entre eux, principalement le croisement et la réduction de la population [302].

Il existe plusieurs architectures parallèles différentes. En fonction du nombre d'instructions pouvant être exécutées simultanément et du nombre de flux de données sur lesquels ces instructions peuvent fonctionner, une taxonomie bien connue proposée par Flynn en 1966 permet de regrouper les architectures en quatre groupes : Instruction unique/Données uniques (SISD), Instruction unique/Données multiples (SIMD), Instruction multiple/Données uniques (MISD), Instruction multiple/Données multiples (MIMD) [302].

D'une part, cette classification est universellement acceptée, utilise des notations courtes et permet une classification facile d'un système. En revanche, avec cette classification, la comparaison des différents systèmes est limitée, les interconnexions, les entrées/sorties, les mémoires ne sont pas prises en compte dans le diagramme. Il est donc rare de classer les architectures complexes modernes dans l'une des catégories de Flynn [303].

En effet, la plupart des architectures modernes implémentent plusieurs de ces paradigmes, comme les processeurs multi-cœurs modernes. Ces processeurs peuvent être de type MIMD, en raison de leur implémentation multi-cœur, mais peuvent aussi être de type SIMD en raison de leurs pipelines d'instructions et de leur jeu d'instructions vectorielles [302, 303]. Puisqu'un ordinateur parallèle MIMD peut être caractérisé par un ensemble de processeurs et de mémoire partagée ou de modules mémoire communicants via un réseau d'interconnexion, une autre façon de classer les architectures parallèles a été proposée en considérant l'organisation, la mémoire et la communication inter-processus. Les plus courantes aujourd'hui sont l'accès mémoire uniforme (UMA), l'accès mémoire non uniforme (NUMA) et l'accès mémoire sans distance (NORMA) [303–305]. D'autres classifications sont possibles, comme celle basée sur l'agencement des mémoires, la communication et le type de parallélisme. Ces dernières années, plusieurs architectures parallèles ont vu le jour. Ces nouvelles architectures peuvent être classées en quatre groupes principaux [306] :

- i. Système sur puce (Soc) / Réseau sur puce (NoC) / Système sur puce multiprocesseur (MPSoC),
- ii. Architectures FPGA,
- iii. Plateformes informatiques distribuées,
 - Grappes d'ordinateurs,
 - Grille,
 - Environnements de cloud computing.
- iv. Ordinateurs multiprocesseurs
 - Ordinateurs multicœurs,
 - Ordinateurs multinoyaux,
 - GPU.

Dans notre cas, nous étions intéressés par l'application des GPUs, car l'implémentation d'algorithmes de régression et d'autres modèles d'apprentissage automatique sur GPU permet d'accélérer significativement les étapes d'entraînement et de prédiction. Étant donné que ces algorithmes ont été souvent des opérations matricielles et généralement intensives, les GPU, avec leur capacité à gérer des calculs en parallèle sur de grands ensembles de données, offrent un avantage significatif par rapport aux CPU et à d'autres architectures parallèles comme les FPGA ou les ASIC (Application-Specific Integrated Circuits). De plus, le choix des GPUs pour l'accélération des MOEAs et des algorithmes de régression repose sur leur architecture massivement parallèle, leur écosystème logiciel riche, leur scalabilité, leur rapport performance/prix attractif, et leur large adoption. Ces facteurs combinés rendent les GPUs particulièrement bien adaptés pour répondre aux besoins de calcul intensifs rencontrés dans des applications comme l'optimisation de déploiement de WSN en PA.

3.9 Graphic Processing Unit (GPU)

Depuis quelques années, les CPU commencent à montrer leurs limites technologiques en terme d'architecture et de vitesse. Leurs architectures multi-cœurs (plusieurs cœurs de calcul au sein d'un même microprocesseur), leur permet de fournir une puissance de calcul toujours plus élevée. Mais cette architecture a une limite qui est liée au temps de latence relativement long lors du transfert des informations entre la mémoire et le microprocesseur. Autrement dit, la bande passante ou la quantité d'informations transférées par seconde, n'est pas suffisante et est un facteur très limitant pour les performances des CPU. Contrairement aux GPUs actuels offrent une très grande puissance de calcul, à un coût relativement réduit. En effet, l'évolution de ces processeurs est telle que leur puissance dépasse largement celle des CPU [307].

3.9.1 Architecture de base du GPU

L'architecture des GPUs est très spécialisée (Figure 3.21) et la plupart des opérations sont figées. Les données sont traitées massivement en parallèle en appliquant la même opération, dans une unité dédiée, à toutes les données (modèle ou/ou noyau ou stream processing). C'est cette spécificité qui permet au GPU d'avoir une puissance de calcul très élevée. L'architecture d'un GPU se compose traditionnellement d'un pipeline de calcul.

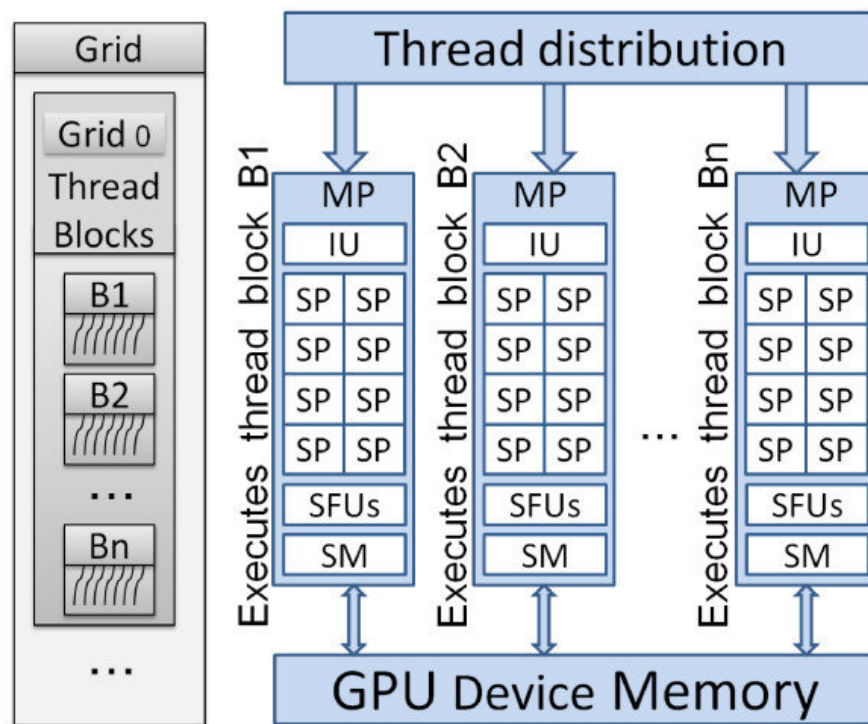


FIGURE 3.21 – Architecture de base d'un GPU [308].

Le processeur de sommets (Vertex Processor) applique des transformations aux points 3D (position, éclairage, etc.), qui sont ensuite assemblés en triangles. Ces triangles sont ensuite convertis en objets affichables à l'écran grâce au processus de rasterisation. Le processeur de fragments applique la texture sur les objets et calcule le rendu final. La puissance de calcul vient de la spécificité des traitements effectués. Les opérations de contrôle sont très réduites et chaque type d'opération est réalisé dans une unité dédiée. Dans ce modèle d'architecture, le processeur de sommets et le processeur de fragments sont programmables. Initialement introduits pour créer des

effets de rendu personnalisés, les possibilités de programmation entendent l'utilisation des processeurs graphiques à d'autres types de calcul. Les programmes, appelés shaders, sont donc appliqués sur les sommets et les pixels[309].

3.10 Parallélisations du modèle de déploiement

3.10.1 Étude de l'existant

Les MOEAs ont prouvé leur efficacité dans l'optimisation du déploiement des nœuds dans les WSN pour une application de surveillance des sols. Cependant, la nature de ces systèmes qui se repose sur la recherche globale et qui intervient plusieurs paramètres et traite d'énormes quantités de données simultanément peut causer un NP-hard pour la consommation de ressources CPU et/ou de mémoire, ce qui force le temps d'exécution à devenir de plus en plus important et peut avoir une mauvaise influence sur les qualités des solutions [199]. Bien entendu que les MOEAs ont une architecture qui permet de calculer et de traiter plusieurs données et instructions simultanément à chaque étape de l'algorithme. Il est remarquable que le parallélisme dans les MOEAs soit très net à de nombreuses étapes, en particulier dans l'étape d'évaluation qui est l'étape la plus consommatrice d'un MOEA, où l'évaluation de tous les individus peut se faire en même temps car ces solutions sont indépendantes l'une des autres. De plus, on ne peut pas appliquer le parallélisme sur une seule étape des MOEAs, qui est l'étape de remplacement, où les parents ne peuvent pas apparaître plusieurs fois dans les générations, afin de garder la diversité dans la population pour ne pas réduire le pouvoir d'exploration de l'algorithme [310]. Ainsi, afin de bénéficier de cette fonctionnalité du parallélisme des MOEAs, les chercheurs se sont inspirés d'utiliser des plates-formes matérielles comme celles du calcul distribué, pour optimiser le temps d'exécution des MOEAs. Dans ce qui suit, nous allons présenter une étude [2] que nous avons faite sur 71 travaux d'optimisation du temps de calcul des EAs et les types de plateformes matérielles utilisées. Nous avons constaté que sur 71 études, seuls 1,40 % utilisaient des systèmes multiprocesseurs sur puce, 8,45 % utilisaient des réseaux prédéfinis programmables sur le terrain (FPGA) et avec le même pourcentage de 8,45 %, ils utilisaient des environnements de Cloud Computing, 12,67 % ont

utilisé les plates-formes Grid, 28,16 % ont utilisé les systèmes Multicore Manycore et Graphics Processing Unit) GPU, et les ordinateurs en cluster (HPC) et les processeur massivement parallèle étaient les plates-formes les plus utilisées avec un pourcentage de 42,25 %. Comme le montre la Figure 3.22.

Plateformes matérielles pour accélérer les MOEAs

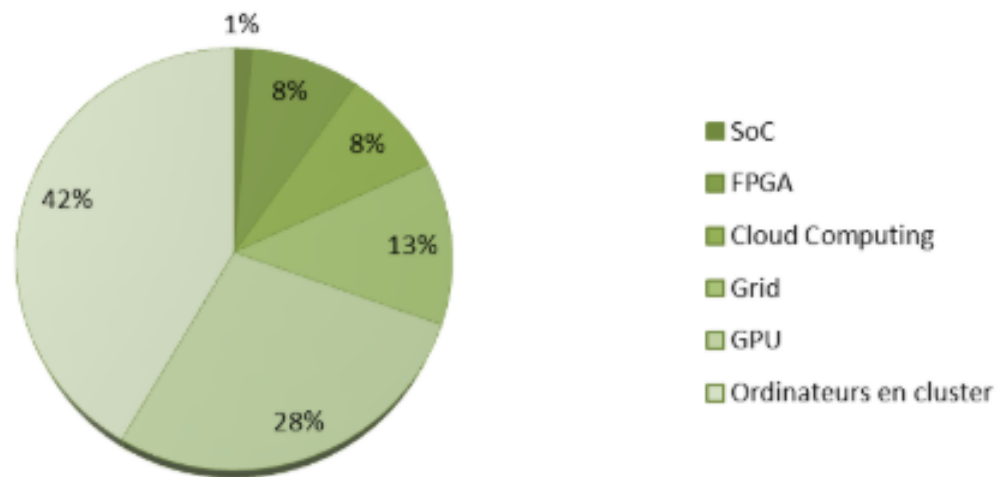


FIGURE 3.22 – Pourcentage de l'utilisation des plates-formes matérielles de calcul distribué pour l'optimisation du temps d'exécution des MOEAs [2].

3.10.2 Architecture parallèle proposée

Dans notre cas d'étude, d'après les essais de notre modèle propose pour le déploiement des nœuds pour la parcelle agricole de grandes tailles et selon les 5 contraintes proposant, l'étape de l'évaluation multi-objectif été l'étape la plus gourmande en temps de calcul. De ce fait, pour remédier au problème de latence d'exécution et en exploitant la capacité d'appliquer le calcul parallèle dans le processus des algorithmes évolutionnaires, nous avons proposé d'appliquer le calcul distribué sur notre modèle de déploiement. En analysant les 5 contraintes de déploiement que nous proposons avant l'application de la surveillance du sol, on peut constater que ces contraintes sont indépendantes de l'une des autres. En conséquence, nous proposons d'évaluer les 5 contraintes en parallèle comme un premier niveau de parallélisations.

Par la suite, pour chaque contrainte de déploiement, on peut évaluer les individus des différentes populations en parallèle ce qui va nous permettre d'accélérer plus l'exécution de notre modèle de déploiement comme le montre l'architecture proposée sur la Figure 3.23.

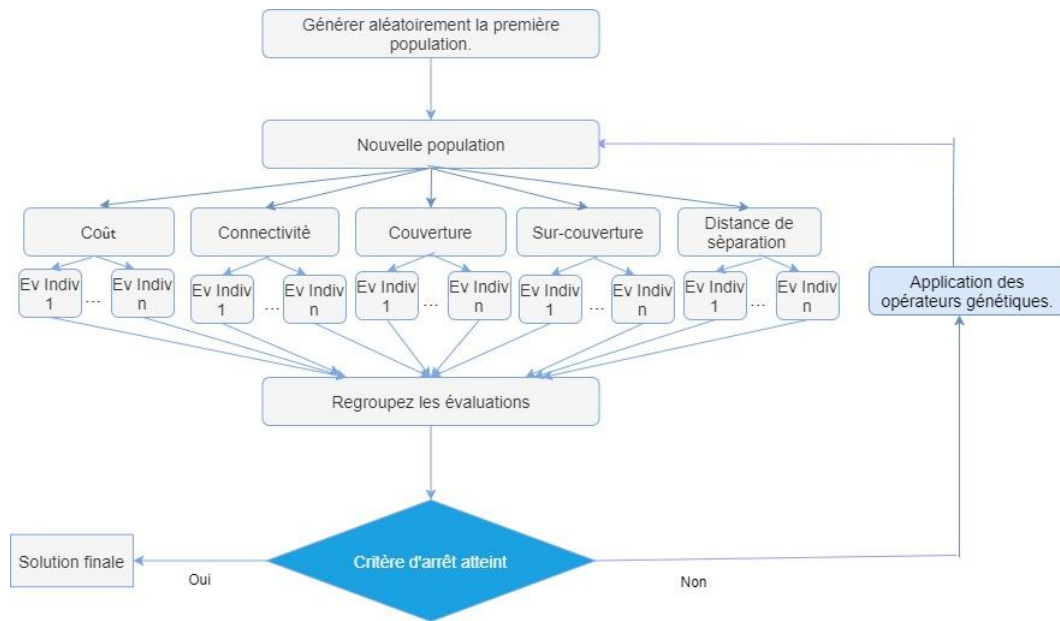


FIGURE 3.23 – Architecture proposée de la solution matérielle.

Le tableau 3.6 présente les résultats de l'algorithme SPEA II exécuté sur deux types de matériel : CPU et GPU. le GPU utilisé c'est Le NVIDIA Tesla T4 disponible gratuitement sur Google Colab qui est un GPU polyvalent, bien adapté pour les travaux de recherche, l'optimisation des algorithmes, et les tâches d'apprentissage profond. Ses cœurs CUDA et Tensor, couplés à une mémoire GDDR6 de 16 Go, permettent d'accélérer efficacement les calculs intensifs. Les performances sont mesurées à travers divers critères tels que le temps d'exécution, le nombre de capteurs, la couverture, la sur-couverture, la connectivité et la distance.

3.10.2.1 Temps d'exécution

L'exécution sur GPU affiche un temps d'exécution de 1689.15 secondes, nettement inférieur aux 2342.59 secondes nécessaires pour l'exécution sur CPU. Cela représente une réduction d'environ 28% du temps d'exécution, démontrant l'efficacité accrue du GPU pour les calculs intensifs et permettant ainsi une optimisation plus rapide des algorithmes.

TABLE 3.6 – Comparaison entre l'exécution du SPEA II sur deux types de matériel : CPU et GPU. pour une taille de parcelle de $400 m^2$.

MOGAs	Temps d'exécution (s)	Nombre des cap- teurs(%)	couverture (%)	Sur- couverture (%)	Connectivité (%)	Distance (m)
SPEA II (CPU)	2342.59	30.5	99.75	77.25	100.0	151859.30
SPEA II (GPU)	1689.15	31.5	100.0	77.0	100.0	158833.64

3.10.2.2 Nombre de capteurs

Le nombre de capteurs utilisés par le SPEA II est légèrement supérieur lorsque l'algorithme est exécuté sur GPU, avec une moyenne de 31.5 capteurs, contre 30.5 pour l'exécution sur CPU. Cette augmentation peut contribuer à améliorer la couverture du réseau, tout en maintenant des performances optimales.

3.10.2.3 Couverture

La couverture des capteurs atteint presque 100% dans les deux cas. Toutefois, le GPU parvient à une couverture parfaite de 100%, comparée à 99.75% pour le CPU. Une couverture complète est essentielle pour garantir que tous les points d'intérêt dans le réseau sont surveillés efficacement.

3.10.2.4 Sur-couverture

La sur-couverture, qui représente la redondance dans la couverture des zones par les capteurs, est légèrement plus faible avec le GPU (77.0%) par rapport au CPU (77.25%). Une sur-couverture moindre peut être bénéfique pour économiser des ressources sans compromettre la qualité de la surveillance.

3.10.2.5 Connectivité

Les deux configurations, CPU et GPU, garantissent une connectivité complète de 100%. Cela est crucial pour s'assurer que toutes les données capturées par les capteurs sont transmises efficacement à travers le réseau, permettant ainsi une communication fluide et sans perte de données.

3.10.2.6 Distance

La distance totale parcourue ou mesurée par les capteurs est légèrement plus élevée avec le GPU (158833.64 unités) par rapport au CPU (151859.30 unités). Cela peut indiquer que la configuration des capteurs maximisant la couverture et la connectivité implique des distances légèrement plus longues. Cependant, cette augmentation de la distance peut être justifiée par l'amélioration des autres critères de performance. En conclusion, l'exécution de l'algorithme SPEA II sur GPU présente plusieurs avantages par rapport à son exécution sur CPU :

- Temps d'exécution réduit : Le GPU permet une optimisation plus rapide, réduisant le temps d'exécution d'environ 28%.
- Amélioration de la couverture : Le GPU atteint une couverture parfaite de 100%, assurant que toutes les zones d'intérêt sont surveillées.
- Nombre de capteurs légèrement augmenté : L'utilisation de plus de capteurs peut contribuer à une meilleure couverture et redondance.
- Sur-couverture réduite : Le GPU parvient à maintenir une couverture optimale avec moins de redondance, économisant ainsi des ressources.
- Connectivité parfaite : Les deux configurations garantissent une connectivité complète, mais le GPU maintient cette connectivité avec une optimisation supplémentaire.

En résumé, l'utilisation de GPU pour l'exécution de l'algorithme SPEA II dans l'optimisation du déploiement des capteurs dans un WSN est avantageuse en termes de rapidité, couverture et dans l'utilisation efficace des ressources.

3.11 Conclusion

L'accélération des MOEAs est cruciale pour leur application à des problèmes de grande envergure nécessitant des solutions optimales en temps raisonnable. Dans ce chapitre, nous avons exploré deux méthodes principales d'accélération : l'hybridation logicielle et l'accélération matérielle.

L'hybridation logicielle des MOEAs avec des algorithmes de régression comme MLP, KNN, DT et les RF s'est révélée particulièrement efficace pour l'approximation des valeurs de fitness. Cette approche permet de réduire considérablement le nombre d'évaluations de fitness nécessaires, accélérant ainsi le processus d'optimisation sans compromettre la qualité des solutions obtenues.

Par ailleurs, l'utilisation des GPUs pour la parallélisation des MOEAs offre une augmentation significative de la vitesse de calcul en répartissant les tâches sur de multiples cœurs de traitement. Cependant, cette approche nécessite souvent une infrastructure matérielle spécifique et peut être moins flexible que l'hybridation logicielle.

En comparant les deux méthodes, nous avons constaté que l'approche logicielle propose une solution plus flexible et généralement plus efficace en termes de temps de calcul et de ressources nécessaires. L'hybridation avec des algorithmes de régression permet non seulement d'accélérer les MOEAs, mais aussi de conserver une grande précision dans les résultats. Par conséquent, pour de nombreux cas pratiques, l'accélération logicielle s'avère être une solution supérieure à l'accélération matérielle.

Conclusion générale

Conclusion générale et perspectives

Pour conclure cette thèse, nous avons proposé une modélisation innovante pour résoudre le problème complexe du déploiement de nœuds dans un Réseau de Capteurs Sans Fil (WSN) en tenant compte des contraintes essentielles telles que le coût, la couverture, la sur-couverture, la connectivité, et la distance de séparation entre les nœuds. Cette approche a conduit au développement d'un outil puissant d'aide à la conception de WSN, capable de répondre efficacement aux exigences spécifiques d'une application donnée.

L'application des algorithmes NSGA-II et SPEA-II pour l'optimisation du déploiement des nœuds a prouvé leur efficacité dans la résolution de ces problèmes multi-objectifs. Les résultats de notre étude indiquent que le NSGA II se distingue par sa rapidité, étant légèrement plus performant en termes de temps de convergence. Cependant, le SPEA II a démontré sa capacité à fournir des solutions de meilleure qualité, ce qui est crucial pour certaines applications où la précision et la fiabilité sont primordiales.

L'un des aspects les plus innovants de notre recherche réside dans l'accélération du SPEA-II grâce à son hybridation avec les algorithmes de régression qui sont le Perceptrons Multicouches (MLP), le k-plus Proches Voisins (KNN), les arbres de Décision (DT) et le Forêts d'Arbres Décisionnels (RF) . Cette hybridation a permis de réduire significativement le temps nécessaire pour déterminer un déploiement optimal des nœuds, particulièrement pour une parcelle agricole de 400 m². Plus précisément, nous avons pu économiser plus de 32 minutes, le modèle RF-SPEA II étant 7 fois plus rapide que le modèle SPEA II brut. Cette amélioration substantielle souligne l'importance de l'intégration des techniques de machine learning dans l'optimisation multi-objectifs, surtout dans des scénarios où le temps de calcul est un facteur limitant.

De plus, L'utilisation des GPUs pour la parallélisation des MOEAs permet d'augmenter considérablement la vitesse de calcul en répartissant les tâches sur de nombreux cœurs de traitement. Cependant, cette approche requiert souvent une infrastructure matérielle spécifique et peut offrir moins de flexibilité que les solutions logicielles hybrides.

En comparant ces deux méthodes, nous avons observé que l'approche logicielle est généralement plus flexible et plus efficace en termes de temps de calcul et de consommation de ressources. L'hybridation avec des algorithmes de régression permet non seulement d'accélérer les MOEAs, mais aussi de maintenir une grande précision des résultats. Ainsi, dans de nombreux cas pratiques, l'accélération logicielle s'avère être une solution préférable à l'accélération matérielle.

Perspectives pour les Travaux Futurs Pour les travaux futurs, nous envisageons plusieurs axes d'amélioration et d'extension de cette recherche :

— **Parallélisation du Modèle :**

Nous proposons d'appliquer l'architecture de parallélisation du modèle sur d'autres infrastructure matérielle, comme les ordinateurs en clusters (HPC). Cette étape vise à mieux explorer le potentiel des architectures parallèles pour accélérer encore davantage notre solution et la rendre compétitive pour des environnements de calcul à haute performance. Nous prévoyons de comparer les résultats obtenus avec ceux obtenues par l'application des GPUs pour identifier les améliorations possibles en termes de temps de calcul et de qualité des solutions.

— **Développement d'une Interface Utilisateur Graphique (GUI) :**

Une interface graphique conviviale serait un ajout précieux à notre application, permettant une interaction plus intuitive et facilitée pour les utilisateurs. Cette interface offrirait la possibilité :

— **D'introduire les dimensions de la parcelle :** Les utilisateurs pourront spécifier la taille et la forme de la zone à surveiller, ce qui est crucial pour adapter le déploiement des capteurs aux besoins spécifiques du terrain.

— **D'introduire les rayons de communication et de détection :** Selon les techniques et les capteurs utilisés, il sera possible de paramétrer ces

éléments pour optimiser la couverture et la connectivité du réseau.

- **D’ajuster les poids des critères** : En fonction des préférences des utilisateurs et des objectifs spécifiques de l’application, les poids des différents critères de performance peuvent être ajustés pour personnaliser la solution de déploiement.

En résumé, cette thèse ouvre la voie à de nouvelles perspectives dans le domaine de l’optimisation multi-objectifs appliquée aux WSN. Les avancées proposées ici, tant sur le plan méthodologique que technologique, posent les bases pour des recherches futures qui visent à pousser plus loin les limites de ce domaine en pleine évolution. L’intégration de techniques avancées de machine learning et d’architectures de calcul parallèle promet d’apporter des améliorations significatives aux systèmes WSN, les rendant plus efficaces, adaptatifs, et capables de répondre aux défis complexes des applications modernes.

Bibliographie

- [1] T. Tušar and B. Filipič, “Visualization of pareto front approximations in evolutionary multiobjective optimization : A critical review and the prosecution method,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 225–245, 2014.
- [2] S. Ferhat Taleb, N. E.-H. Benalia, and R. Sadoun, “Evolutionary algorithm applications for iots dedicated to precise irrigation systems : state of the art,” *Evolutionary Intelligence*, vol. 16, no. 2, pp. 383–400, 2023.
- [3] A. Boretti and L. Rosa, “Reassessing the projections of the world water development report,” *NPJ Clean Water*, vol. 2, no. 1, p. 15, 2019.
- [4] P. H. Glaser, J. A. Janssens, and D. I. Siegel, “The response of vegetation to chemical and hydrological gradients in the lost river peatland, northern minnesota,” *The Journal of Ecology*, pp. 1021–1048, 1990.
- [5] L. Karthikeyan, I. Chawla, and A. K. Mishra, “A review of remote sensing applications in agriculture for food security : Crop growth and yield, irrigation, and crop losses,” *Journal of Hydrology*, vol. 586, p. 124905, 2020.
- [6] V. Puri, A. Nayyar, and L. Raja, “Agriculture drones : A modern breakthrough in precision agriculture,” *Journal of Statistics and Management Systems*, vol. 20, no. 4, pp. 507–518, 2017.
- [7] D. Thakur, Y. Kumar, A. Kumar, and P. K. Singh, “Applicability of wireless sensor networks in precision agriculture : A review,” *Wireless Personal Communications*, vol. 107, pp. 471–512, 2019.

- [8] A. Khattab, A. Abdelgawad, and K. Yelmarthi, "Design and implementation of a cloud-based iot scheme for precision agriculture," in *2016 28th international conference on microelectronics (ICM)*. IEEE, 2016, pp. 201–204.
- [9] P. Zwaenepoel and J.-M. Le Bars, "L'agriculture de précision," *Ingénieries eau-agriculture-territoires*, no. 12, pp. p–67, 1997.
- [10] F. J. Pierce and P. Nowak, "Aspects of precision agriculture," *Advances in agronomy*, vol. 67, pp. 1–85, 1999.
- [11] T. Angelopoulou, N. Tziolas, A. Balafoutis, G. Zalidis, and D. Bochtis, "Remote sensing techniques for soil organic carbon estimation : A review," *Remote Sensing*, vol. 11, no. 6, p. 676, 2019.
- [12] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
- [13] S. M. Islam, Y. K. Gaihre, J. C. Biswas, M. S. Jahan, U. Singh, S. K. Adhikary, and M. A. Saleque, "Different nitrogen rates and methods of application for dry season rice cultivation with alternate wetting and drying irrigation : Fate of nitrogen and grain yield," *Agricultural Water Management*, vol. 196, pp. 144–153, 2018.
- [14] O. Adeyemi, I. Grove, S. Peets, and T. Norton, "Advanced monitoring and management systems for improving sustainability in precision irrigation," *Sustainability*, vol. 9, no. 3, p. 353, 2017.
- [15] D. N. Baker, J. R. Lambert, and J. M. McKinion, "Gossym : A simulator of cotton crop growth and yield," Agricultural Experiment Station, South Carolina, USA, Technical bulletin, 1983.
- [16] F. Shah and W. Wu, "Soil and crop management strategies to ensure higher crop productivity within sustainable environments," *Sustainability*, vol. 11, no. 5, p. 1485, 2019.

- [17] L. Zhou, N. Chen, Z. Chen, and C. Xing, “Roscc : an efficient remote sensing observation-sharing method based on cloud computing for soil moisture mapping in precision agriculture,” *IEEE Journal of selected topics in applied earth observations and remote sensing*, vol. 9, no. 12, pp. 5588–5598, 2016.
- [18] Y. Kang, M. Özdoğan, S. C. Zipper, M. O. Román, J. Walker, S. Y. Hong, M. Marshall, V. Magliulo, J. Moreno, L. Alonso *et al.*, “How universal is the relationship between remotely sensed vegetation indices and crop leaf area index ? a global assessment,” *Remote sensing*, vol. 8, no. 7, p. 597, 2016.
- [19] E. Salas and S. Subburayalu, “Modified shape index for object-based random forest image classification of agricultural systems using airborne hyperspectral datasets,” *PLoS ONE*, vol. 14, no. 3, p. e0213356, 2019.
- [20] J. Yue, G. Yang, C. Li, Z. Li, Y. Wang, H. Feng, and B. Xu, “Estimation of winter wheat above-ground biomass using unmanned aerial vehicle-based snapshot hyperspectral sensor and crop height improved models,” *Remote Sensing*, vol. 9, no. 7, p. 708, 2017.
- [21] D. Burles, *Dimensions of need : an atlas of food and agriculture*. Food and Agriculture Organization of the United Nations, 1995.
- [22] Sols-Mesures, “Planimètre foliaire am350,” <http://www.sols-mesures.com/?product=planimetre-foliaire-am350>, 2024, consulté le 3 novembre 2024.
- [23] A. E. Ercin and A. Y. Hoekstra, “Water footprint scenarios for 2050 : A global analysis,” *Environment International*, vol. 64, pp. 71–82, 2014.
- [24] Z. Huang, M. Hejazi, Q. Tang, C. R. Vernon, Y. Liu, M. Chen, and K. Calvin, “Global agricultural green and blue water consumption under future climate and land use changes,” *Journal of Hydrology*, vol. 574, pp. 242–256, 2019.
- [25] J. Wang, W. Niu, L. Guo, B. Liang, and Y. Li, “Suitable buried depth of drip irrigation improving yield and quality of tomato in greenhouse,” *Transactions of the Chinese Society of Agricultural Engineering*, vol. 33, no. 20, pp. 90–97, 2017.

- [26] A. Mouradi, Z. A. Yacine, and A. El Harti, “Study of the technical performance of localized irrigation and its environmental and agro economic impact in the first areas of collective reconversion at the irrigated perimeter of the tadla beni moussa perimeter of the west morocco,” in *E3S Web of Conferences*, vol. 37. EDP Sciences, 2018, p. 01009.
- [27] F. Zazueta, A. Smajstrla, and G. Clark, *Irrigation system controllers*. Institute of Food and Agriculture Science, University of Florida (AGE-32), 1994.
- [28] Wikifarmer, “Conservation des sols et de l’eau,” <https://wikifarmer.com/fr/category/conservation-des-sols-et-de-leau/>, 2024, consulté le 3 novembre 2024.
- [29] F. M. Rhoads and C. D. Yonts, “Irrigation scheduling for corn-why and how,” *National Corn Handbook*, vol. 20, 1991.
- [30] J. E. Fernandez, “Plant-based methods for irrigation scheduling of woody crops,” *Horticulturae*, vol. 3, no. 2, p. 35, 2017.
- [31] M. V. C. Caya, J. B. G. Ibarra, G. O. Avendano, D. J. D. A. Felipe, J. A. V. Fernando, J. M. T. Galvez, and Z. Sauli, “Evapotranspiration based irrigation system using raspberry pi for capsicum annuum ‘bell pepper’ plant nursery,” *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 1-14, pp. 21–24, 2018.
- [32] S. ShaanandS. J. Maas, “Index of soil moisture using raw landsat imagedigital count data in texas—2372, 2015.
- [33] J. M. Norman and G. Campbell, “Application of a plant-environment model to problems in irrigation,” in *Advances in Irrigation*. Elsevier, 1983, vol. 2, pp. 155–188.
- [34] L. S. R. Mechs, M. U. B. Dias, W. Pragithmukar, and A. L. Kulasekera, “A mobile robot based watering system for smart lawn maintenance,” in *17th International Conference on Control, Automation and Systems (ICCAS)*, 2017, p. 2017.
- [35] B. Khelifa, D. Amel, B. Amel, C. Mohamed, and B. Tarek, “Smart irrigation using internet of things,” in *2015 Fourth International Conference on Future Generation Communication Technology (FGCT)*. IEEE, 2015, pp. 1–5.

- [36] J. Melkonian and H. van Es, “Adapt-n : Adaptive nitrogen management for maize using high resolution climate data and model simulations,” in *Proceedings of the 9th International Conference on Precision Agriculture*, Denver, CO, USA, 2008.
- [37] M. Ali, A. Al-Ani, D. Eamus, and D. Tan, “Leaf nitrogen determination using non-destructive techniques—a review,” *Journal of Plant Nutrition*, vol. 40, no. 7, pp. 928–953, 2017.
- [38] UFArevue, “Quelle quantité d’eau et à quel moment,” <https://www.ufarevue.ch/fr/e/production-vegetale/quelle-quantite-d-eau-et-a-quel-moment>, 2024, consulté le 3 novembre 2024.
- [39] S. Marino and A. Alvino, “Hyperspectral vegetation indices for predicting onion (*allium cepa* l.) yield spatial variability,” *Computers and Electronics in Agriculture*, vol. 116, pp. 109–117, 2015.
- [40] G. Blasch, D. Spengler, C. Hohmann, C. Neumann, S. Itzerott, and H. Kaufmann, “Multitemporal soil pattern analysis with multispectral remote sensing data at the field scale,” *Computers and Electronics in Agriculture*, vol. 113, pp. 1–13, 2015.
- [41] D. D. K. Rathinam, D. Surendran, A. Shilpa, A. S. Grace, and J. Sherin, “Modern agriculture using wireless sensor network (wsn),” in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. IEEE, 2019, pp. 515–519.
- [42] R. Ehsani and J. Maja, “The rise of small uavs in precision agriculture,” *Resource Magazine*, vol. 20, pp. 18–19, 2013.
- [43] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep neural networks based recognition of plant diseases by leaf image classification,” *Computational Intelligence and Neuroscience*, vol. 2016, 2016.
- [44] S. Di Gennaro, E. Battiston, S. Di Marco, O. Facini, A. Matese, M. Nocentini, A. Palliotti, and L. Mugnai, “Unmanned aerial vehicle (uav)-based remote sensing to monitor grapevine leaf stripe disease within a vineyard affected by esca complex,” *Phytopathologia Mediterranea*, vol. 55, no. 2, pp. 262–275, 2016.

- [45] N. Srivastav, G. Chopra, P. Jain, and B. Khatter, "Pest monitor and control system using wsn with special reference to acoustic device," in *ICEEE*, India, 2013.
- [46] Sky-Dimensions, "Thermographie par drone," <https://www.sky-dimensions.fr/thermographie-par-drone/>, 2024, consulté le 3 novembre 2024.
- [47] A. Monteiro, S. Santos, and P. Gonçalves, "Precision agriculture for crop and livestock farming—brief review," *Animals*, vol. 11, no. 8, p. 2345, 2021.
- [48] N. Hostiou, J. Fagon, S. Chauvat, A. Turlot, F. Kling-Eveillard, X. Boivin, and C. Alain, "Impact of precision livestock farming on work and human-animal interactions on dairy farms. a review," *Biotechnologie, Agronomie, Société et Environnement*, vol. 21, pp. 268–275, 2017.
- [49] T. Banhazi, L. Babinszky, V. Halas, and M. Tschärke, "Precision livestock farming : Precision feeding technologies and sustainable livestock production," *International Journal of Agricultural and Biological Engineering*, vol. 5, no. 3, pp. 54–61, 2012.
- [50] E. Sales-Baptista, M. Ferraz-de Oliveira, J. Lopes de Castro, L. Rato, and M. Canceleda d'Abreu, "Pastoreio de precisão : Monitorizar o comportamento dos animais para adaptar a oferta à procura," in *Revista Portuguesa de Zootecnia*. Vila Real, Portugal : Associação Portuguesa de Engenharia Zootécnica (APEZ), 2019, pp. 121–128.
- [51] V. Hertem, "Objective sustainability assessment by precision livestock farming," *Agriculture, Food Policy, Environment, Economics Policy*, 2018.
- [52] D. Lovarelli, J. Bacenetti, and M. Guarino, "A review on dairy cattle farming : Is precision livestock farming the compromise for an environmental, economic and social sustainable production ?" *Journal of Cleaner Production*, vol. 262, p. 121409, 2020.
- [53] S. Neethirajan, "The role of sensors, big data and machine learning in modern animal farming," *Sensing and Bio-Sensing Research*, vol. 29, p. 100367, 2020.
- [54] Web-Agri, "Les colliers medria mesurent les temps d'ingestion et la rumination," <https://www.web-agri.fr/2014/article/104294/les-colliers-medria-mesurent-les-temps-d-ingestion-et-la-rumination>, 2014, consulté le 3 novembre 2024.

- [55] H. Tian, T. Wang, Y. Liu, X. Qiao, and Y. Li, “Computer vision technology in agricultural automation—a review,” *Information Processing in Agriculture*, vol. 7, no. 1, pp. 1–19, 2020.
- [56] E. C. Research*eu, *Precision farming : Sowing the seeds of a new agricultural revolution*. Luxembourg : The Community Research and Development Information Service (CORDIS), 2017.
- [57] J. R. C. J. of the European Commission, P. Zarco-Tejada, N. Hubbard, and P. Loudjani, “Precision agriculture : An opportunity for eu farmers—potential support with the cap 2014–2020,” Agriculture and Rural Development ; Policy Department B : Structural and Cohesion Policies European Union ; European Parliament, Brussels, Belgium, Tech. Rep., 2014.
- [58] A. Boursianis, M. Papadopoulou, P. Diamantoulakis, A. Liopa-Tsakalidi, P. Barouchas, G. Salahas, G. Karagiannidis, S. Wan, and S. Goudos, “Internet of things (iot) and agricultural unmanned aerial vehicles (uavs) in smart farming : A comprehensive review,” *Int. Things*, vol. 100187, 2020.
- [59] A. Ghafar, S. Hajjaj, K. Gsangaya, M. Sultan, M. Mail, and L. Hua, “Design and development of a robot for spraying fertilizers and pesticides for agriculture,” *Mater. Today Proc.*, 2021.
- [60] S. Vougioukas, “Agricultural robotics,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 365–392, 2019.
- [61] H. Griepentrog, M. Norremark, H. Nielsen, and B. Blackmore, “Seed mapping of sugar beet,” *Precis. Agric.*, vol. 6, no. 2, pp. 157–165, 2005.
- [62] T. Utstumo, F. Urdal, A. Brevik, J. Dørum, J. Netland, Overskeid, T. Berge, and J. Gravdahl, “Robotic in-row weed control in vegetables,” *Comput. Electron. Agric.*, vol. 154, pp. 36–45, 2018.
- [63] L’Observateur, “Stratégie génération green : on en est où ?” <https://lobservateur.info/article/106447/economie/strategie-generation-green-on-en-est-ou->, 2024, consulté le 3 novembre 2024.

- [64] T. Bakker, K. Van Asselt, J. Bontsema, J. Müller, and G. Van Straten, “An autonomous weeding robot for organic farming,” in *Field and Service Robotics*, vol. 25. Berlin, Germany : Springer, 2006, pp. 579–590.
- [65] N. Schor, A. Bechar, T. Ignat, A. Dombrovsky, Y. Elad, and S. Berman, “Robotic disease detection in greenhouses : Combined detection of powdery mildew and tomato spotted wilt virus,” *IEEE Robot. Autom. Lett.*, vol. 1, pp. 354–360, 2016.
- [66] N. Schor, S. Berman, A. Dombrovsky, Y. Elad, T. Ignat, and A. Bechar, “Development of a robotic detection system for greenhouse pepper plant diseases,” *Precis. Agric.*, vol. 18, pp. 394–409, 2017.
- [67] J. Katupitiya, “An autonomous seeder for broad acre crops,” in *Proceedings of the American Society of Agricultural and Biological Engineers Annual International Meeting*, vol. 1, 2014, pp. 169–176.
- [68] P. Ruangurai, M. Ekpanyapong, C. Pruetong, and T. Watwai, “Automated three-wheel rice seeding robot operating in dry paddy fields,” *Maejo Int. J. Sci. Technol.*, vol. 9, pp. 403–412, 2015.
- [69] R. Klose, M. Thiel, A. Ruckelshausen, and J. Marquering, “Weedy—a sensor fusion based autonomous field robot for selective weed control,” in *Proceedings of the Conference : Agricultural Engineering—Land-Technik 2008 : Landtechnik regional und International*, 2008, pp. 167–172.
- [70] B. Rey, N. Aleixos, S. Cubero, and J. Blasco, “Xf-rovim. a field robot to detect olive trees infected by xylella fastidiosa using proximal sensing,” vol. 11, 2019, p. 221.
- [71] B. Liu, Z. Hu, Y. Zhao, Y. Bai, and Y. Wang, “Recognition of pyralidae insects using intelligent monitoring autonomous robot vehicle in natural farm scene,” *arXiv preprint arXiv :1903.10827*, 2019.
- [72] F. Dos Santos, H. Sobreira, D. Campos, R. Morais, A. Moreira, and O. Contente, “Towards a reliable monitoring robot for mountain vineyards,” in *Proceedings of the 2015 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2015*, 2015, pp. 37–43.

- [73] T. Mueller-Sim, M. Jenkins, J. Abel, and G. Kantor, "The robotanist : A ground-based agricultural robot for high-throughput crop phenotyping," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2017, pp. 3634–3639.
- [74] P. Sammons, T. Furukawa, and A. Bulgin, "Autonomous pesticide spraying robot for use in a greenhouse," in *Proceedings of the 2005 Australasian Conference on Robotics and Automation*.
- [75] J. Underwood, M. Calleija, Z. Taylor, C. Hung, J. Nieto, R. Fitch, and S. Sukkarieh, "Real-time target detection and steerable spray for vegetable crops," in *Proceedings of the International Conference on Robotics and Automation : Robotics in Agriculture Workshop*.
- [76] T. Pilarski, M. Happold, H. Pangels, M. Ollis, K. Fitzpatrick, and A. Stentz, "The demeter system for automated harvesting," *Auton. Robots*, vol. 13, pp. 9–20, 2002.
- [77] A. Silwal, J. Davidson, M. Karkee, C. Mo, Q. Zhang, and K. Lewis, "E_orttowardsroboticappleharvestinginwashingtonstate," in *Proceedings of the 2016 American Soc*
- [78] P. Shanmugapriya, S. Rathika, T. Ramesh, and P. Janaki, "Applications of remote sensing in agriculture—a review," *Int. J. Curr. Microbiol. Appl. Sci*, vol. 8, no. 01, pp. 2270–2283, 2019.
- [79] S. Khanal, K. Kc, J. Fulton, S. Shearer, and E. Ozkan, "Remote sensing in agriculture—accomplishments, limitations, and opportunities," *Remote Sensing*, vol. 12, no. 22, p. 3783, 2020.
- [80] S. Bhatti, D. Heeren, J. Barker, C. Neale, W. Woldt, M. Maguire, and D. Rudnick, "Site-specific irrigation management in a sub-humid climate using a spatial evapotranspiration model with satellite and airborne imagery," *Agric. Water Manag.*, vol. 230, p. 105950, 2020.
- [81] C. Chang, R. Zhou, O. Kira, S. Marri, J. Skovira, L. Gu, and Y. Sun, "An unmanned aerial system (uas) for concurrent measurements of solar induced chlorophyll fluorescence and hyperspectral reflectance toward improving crop monitoring," *Agric. For. Meteorol.*, vol. 294, pp. 1–15, 2020.

- [82] R. Jackson, "Remote sensing of biotic and abiotic plant stress," *Annual Review of Phytopathology*, vol. 24, pp. 265–286, 1986.
- [83] L. Jarboui, "Méthodes avancées de séparation de sources applicable au mélanges linéaires-quadratiques," Ph.D. dissertation, 11 2017.
- [84] M. Nellis, K. Price, and D. Rundquist, "Remote sensing of cropland agriculture," in *The SAGE Handbook of Remote Sensing*. London, UK : Sage, 2009, vol. 1, pp. 368–380.
- [85] P. Pereira, E. Brevik, M. Muñoz-Rojas, and B. Miller, *Soil Mapping and Process Modeling for Sustainable Land Use Management*. Amsterdam, The Netherlands : Elsevier, 2017.
- [86] A. Calera, I. Campos, A. Osann, G. D’Urso, and M. Menenti, "Remote sensing for crop water management : From et modelling to services for the end users," *Sensors*, vol. 17, p. 1104, 2017.
- [87] S. Di Gennaro, E. Battiston, S. Di Marco, O. Facini, A. Matese, M. Nocentini, A. Palliotti, and L. Mugnai, "Unmanned aerial vehicle (uav)-based remote sensing to monitor grapevine leaf stripe disease within a vineyard affected by esca complex," *Phytopathol. Mediterr.*, vol. 55, pp. 262–275, 2016.
- [88] A. Ali and M. Imran, "Evaluating the potential of red edge position (rep) of hyperspectral remote sensing data for real-time estimation of lai chlorophyll content of kinnow mandarin (*citrus reticulata*) fruit orchards," *Sci. Hortic. Amst.*, vol. 267, p. 109326, 2020.
- [89] S. Gao, Z. Niu, N. Huang, and X. Hou, "Estimating the leaf area index, height and biomass of maize using hj-1 and radarsat-2," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 24, pp. 1–18, 2013.
- [90] C. Leslie, L. Serbina, and H. Miller, *Landsat and Agriculture—Case Studies on the Uses and Benefits of Landsat Imagery in Agricultural Monitoring and Production*. Reston, VA, USA : US Geological Survey Open-File Report, 2017, vol. 1034.

- [91] T. Dong, J. Liu, B. Qian, T. Zhao, Q. Jing, X. Geng, J. Wang, T. Human, and J. Shang, "Estimating winter wheat biomass by assimilating leaf area index derived from MODIS data," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 49, pp. 63–74, 2016.
- [92] P. Worsley and J. Bowler, "Assessing flood damage using spot and noaa avhrr data," *Geospat. Inf. Agric.*, pp. 2–7, 2001, available online : <http://www.regional.org.au/au/gia/12/397worsley.htm#TopOfPage> (accessed on 23 September 2020).
- [93] P. Mondal and M. Basu, "Adoption of precision agriculture technologies in india and in some developing countries : Scope, present status and strategies," *Prog. Nat. Sci.*, vol. 19, pp. 659–666, 2009.
- [94] K. Koenig, B. Höfle, M. Hämmerle, T. Jarmer, B. Siegmann, and H. Lilienthal, "Comparative classification analysis of post-harvest growth detection from terrestrial lidar point clouds in precision agriculture," *ISPRS J. Photogramm. Remote Sens.*, vol. 104, pp. 112–125, 2015.
- [95] E. Omran, "Remote estimation of vegetation parameters using narrow band sensor for precision agriculture in arid environment," *Egypt. J. Soil Sci.*, vol. 58, pp. 73–92, 2018.
- [96] J. Fisher, B. Lee, A. Purdy, G. Halverson, M. Dohlen, K. Cawse-Nicholson, A. Wang, R. Anderson, B. Aragon, M. Arain *et al.*, "Ecostress : Nasa's next generation mission to measure evapotranspiration from the international space station," *Water Resour. Res.*, vol. 56, p. e2019WR026058, 2020.
- [97] H. Santoso, T. Gunawan, R. Jatmiko, W. Darmosarkoro, and B. Minasny, "Mapping and identifying basal stem rot disease in oil palms in north sumatra with quickbird imagery," *Precis. Agric.*, vol. 12, pp. 233–248, 2011.
- [98] L. Yuan, R. Pu, J. Zhang, J. Wang, and H. Yang, "Using high spatial resolution satellite imagery for mapping powdery mildew at a regional scale," *Precis. Agric.*, vol. 17, pp. 332–348, 2016.
- [99] D. Fong, "Internet of things and data analytics handbook," in *Internet of things and data analytics handbook*, 2017, pp. 197–213.

- [100] T. Ojha, S. Misra, and N. Raghuwanshi, "Computers and electronics in agriculture," *Computers and Electronics in Agriculture*, vol. 118, p. 66, 2015.
- [101] N. Meratnia, B. J. van der Zwaag, H. van Dijk, D. Bijwaard, and P. Havinga, "Sensor networks in the low lands," *Sensors (Basel, Switzerland)*, vol. 10, pp. 8504–25, 09 2010.
- [102] P. Rawat, K. Singh, H. Chaouchi, and J. Bonnin, "The journal of supercomputing," *The Journal of Supercomputing*, vol. 68, no. 1, p. 1, 2014.
- [103] P. Baronti, P. Pillai, V. Chook, S. Chessa, A. Gotta, and Y. Hu, "Computer communications," *Computer Communications*, vol. 30, no. 7, p. 1655, 2007.
- [104] P. Gralla, "Precision agriculture yields higher profits, lower risks," *Hewlett Packard Enterprise*, 2018.
- [105] T. Behera, S. Mohapatra, U. Samal, and M. Khan, "Hybrid heterogeneous routing scheme for improved network performance in wsns for animal tracking," *Internet of Things*, vol. 6, p. 100047, 2019.
- [106] L. García, L. Parra, J. Jimenez, M. Parra, J. Lloret, P. Mauri, and P. Lorenz, "Deployment strategies of soil monitoring wsn for precision agriculture irrigation scheduling in rural areas," *Sensors*, vol. 21, no. 5, p. 1693, 2021.
- [107] X. Peng and G. Liu, "Intelligent water-saving irrigation system based on fuzzy control and wireless sensor network," in *2012 Fourth International Conference on Digital Home*. IEEE, 2012, pp. 252–256.
- [108] A. Tzounis, N. Katsoulas, T. Bartzanas, and C. Kittas, "Internet of things in agriculture, recent advances and future challenges," *Biosystems Engineering*, vol. 164, pp. 31–48, 2017.
- [109] N. Katsoulas, K. P. Ferentinos, A. Tzounis, T. Bartzanas, and C. Kittas, "Spatially distributed greenhouse climate control based on wireless sensor network measurements," *Acta Horticulturae*, vol. 1154, pp. 111–120, 2017.

- [110] M. Asikainen, K. Haataja, and P. Toivanen, "Wireless indoor tracking of livestock for behavioral analysis," in *2013 9th International Wireless Communications and Mobile Computing Conference*, 2013, pp. 1833–1838.
- [111] T. F. Khan and D. S. Kumar, "Ambient crop field monitoring for improving context based agricultural by mobile sink in wsn," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1431–1439, 2020.
- [112] G. Portz, J. P. Molin, and J. Jasper, "Active crop sensor to detect variability of nitrogen supply and biomass on sugarcane fields," *Precision Agriculture*, vol. 13, no. 1, pp. 33–44, 2012.
- [113] S. M. A. El-Kader and B. M. M. El-Basioni, "Precision farming solution in egypt using the wireless sensor network technology," *Egyptian Informatics Journal*, vol. 14, no. 3, pp. 221–233, 2013.
- [114] D. Levy, W. K. Coleman, and R. E. Veilleux, "Adaptation of potato to water shortage : irrigation management and enhancement of tolerance to drought and salinity," *American Journal of Potato Research*, vol. 90, no. 2, pp. 186–206, 2013.
- [115] F. Viani, M. Bertolli, M. Salucci, and A. Polo, "Low-cost wireless monitoring and decision support for water saving in agriculture," *IEEE Sensors Journal*, vol. 17, no. 13, pp. 4299–4309, 2017.
- [116] Y. Kim, R. G. Evans, and W. M. Iversen, "Remote sensing and control of an irrigation system using a distributed wireless sensor network," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 7, pp. 1379–1387, 2008.
- [117] R. W. Coates, M. J. Delwiche, A. Broad, and M. Holler, "Wireless sensor network with irrigation valve control," *Computers and Electronics in Agriculture*, vol. 96, pp. 13–22, 2013.
- [118] D. Masseroni, A. Facchi, E. V. Depoli, F. M. Renga, and C. Gandolf, "Irrig-oh : An open-hardware device for soil water potential monitoring and irrigation management," *Irrigation and Drainage*, vol. 65, no. 5, pp. 750–761, 2016.

- [119] R. A. V. Rossel and J. Bouma, "Soil sensing : A new paradigm for agriculture," *Agricultural Systems*, vol. 148, pp. 71–74, 2016.
- [120] M. Knadel, A. Thomsen, K. Schelde, and M. H. Greve, "Soil organic carbon and particle sizes mapping using vis-nir, ec and temperature mobile sensor platform," *Computers and Electronics in Agriculture*, vol. 114, pp. 134–144, 2015.
- [121] T. Jackson, K. Mansfeld, M. Saaf, T. Colman, and P. Romine, "Measuring soil temperature and moisture using wireless mems sensors," *Measurement*, vol. 41, no. 4, pp. 381–390, 2008.
- [122] J. R. Mahan, W. Conaty, J. Neilsen, P. Payton, and S. B. Cox, "Field performance in agricultural settings of a wireless temperature monitoring system based on a low-cost infrared sensor," *Computers and Electronics in Agriculture*, vol. 71, no. 2, pp. 176–181, 2010.
- [123] S. Fellah, "Optimisation multi-objectif appliquée au déploiement et à la performance des réseaux de capteurs sans fil," *Thèses, Université d'Oran1 Ahmed Ben Bella*, 2018.
- [124] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks : a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [125] A. Gutierrez, "Automated title of the paper," in *Proceedings of the Conference Name*. Publisher Name, 2014, pp. 1–10.
- [126] Y. Yu, Q. Wang, and X. Meng, "A survey of sensor network applications," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1025–1056, 2013.
- [127] L. B. Ruiz, J. M. Nogueira, and A. A. F. Loureiro, "Manna : A management architecture for wireless sensor networks," *IEEE Communications Magazine*, vol. 41, no. 2, pp. 116–125, 2003.
- [128] J. Rezazadeh, M. Moradi, and A. S. Ismail, "Mobile wireless sensor networks overview," *International Journal of Computer Communications and Networks*, vol. 2, no. 1, pp. 17–22, 2012.

- [129] C. M. Ramya, M. Shanmugaraj, and R. Prabakaran, "Study on zigbee technology," in *2011 3rd international conference on electronics computer technology*, vol. 6. IEEE, 2011, pp. 297–301.
- [130] G. R. Mendez, M. A. Md Yunus, and S. C. Mukhopadhyay, "A wifi based smart wireless sensor network for an agricultural environment," in *2011 Fifth international conference on sensing technology*. IEEE, 2011, pp. 592–597.
- [131] J. Bjarnason, "Evaluation of bluetooth low energy in agriculture environments," 2017.
- [132] M. S. Kabir *et al.*, "Environmental sensing and remote communication for smart farming : A review," *Precision Agriculture*, vol. 4, pp. 82–94, 2022.
- [133] M. U. Rani *et al.*, "An integration of wireless sensor network through wi-max for agriculture monitoring," in *2014 International Conference on Computer Communication and Informatics*. IEEE, 2014, pp. 1–5.
- [134] P. Kapil and S. Lata, "Review on selecting topologies in zigbee networks," 2016.
- [135] G. Kaur and R. M. Garg, "Energy efficient topologies for wireless sensor networks," *International Journal of Distributed and Parallel Systems (IJDPS)*, vol. 3, no. 5, pp. 179–192, 2012.
- [136] A. Belfkih, B. Sadeg, C. Duvallet, and L. Amanton, "Les bases de données dans les réseaux de capteurs sans fil," *Techniques et Sciences Informatiques*, vol. 33, pp. 739–776, 2014.
- [137] M. Matin, *Wireless sensor network : technology and protocols*. InTech, 2012.
- [138] S. Roundy *et al.*, "Power sources for wireless sensor networks," in *Wireless Sensor Networks : First European Workshop, EWSN 2004, Berlin, Germany, January 19-21, 2004. Proceedings*. Springer, 2004, pp. 1–17.
- [139] M. H. Anisi, A. H. Abdullah, and S. A. Razak, "Energy-efficient data collection in wireless sensor networks," *Wireless Sensor Networks*, vol. 3, pp. 329–333, 2011.

- [140] A. Jangra, Swati, Richa, and Priyanka, “Wireless sensor network (wsn) : Architectural design issues and challenges,” *International Journal on Computer Science and Engineering*, vol. 2, no. 9, pp. 3089–3094, 2010.
- [141] S. Gowrishankar, T. G. Basavaraju, D. H. Manjaiah, and S. K. Sarkar, “Issues in wireless sensor networks,” in *Proceedings of the World Congress on Engineering*, vol. I. IEEE, 2008.
- [142] N. A. A. Aziz, K. A. Aziz, and W. Z. W. Ismail, “Coverage strategies for wireless sensor networks,” *International Journal of Electronics and Communication Engineering*, vol. 3, no. 2, pp. 171–176, 2009.
- [143] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, “Grid coverage for surveillance and target location in distributed sensor networks,” *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1448–1453, 2002.
- [144] I. Khoufi *et al.*, “Survey of deployment algorithms in wireless sensor networks : Coverage and connectivity issues and challenges,” *International Journal of Autonomous and Adaptive Communications Systems (IJAACS)*, vol. 10, no. 4, pp. 341–390, 2017.
- [145] T. M. Megha and P. Mohan, “Performance analysis using time splitter method for power management in wireless sensor network,” in *2021 2nd International Conference for Emerging Technology (INCET)*. IEEE, 2021, pp. 1–4.
- [146] J. M. Keenan and A. J. Motley, “Radio coverage in buildings,” *British Telecom Technology Journal*, vol. 8, no. 1, pp. 19–24, 1990.
- [147] J. Roselin, P. Latha, and S. Benitta, “Maximizing the wireless sensor networks lifetime through energy efficient connected coverage,” *Ad Hoc Networks*, vol. 62, pp. 1–10, 2017.
- [148] Y. Zhang *et al.*, “Ubiquitous wsn for healthcare : Recent advances and future prospects,” *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 311–318, 2014.
- [149] N. Nejah and A. Kachouri, “Etude et proposition d’une méthode de géolocalisation et de suivi de patients en milieu médicalisé,” 11 2019.

- [150] N. de l'auteur (ou liste des auteurs), "Titre du chapitre," in *Titre du livre*. Springer, 2024, p. Pages du chapitre. [Online]. Available : https://link.springer.com/chapter/10.1007/978-3-031-51195-0_1
- [151] E. Kanagaraj *et al.*, "Cloud-based remote environmental monitoring system with distributed wsn weather stations," in *2015 IEEE SENSORS*. IEEE, 2015, pp. 1–4.
- [152] H. Ghayvat *et al.*, "Wsn-and iot-based smart homes and their extension to smart buildings," *Sensors*, vol. 15, no. 5, pp. 10 350–10 379, 2015.
- [153] N. E.-H. Benalia, I. Si Hadj Mohand, S. Ferhattaleb, R. Sadoun, and A. Bentrach, "Moea-deploywsn-sb : Three variants of multi-objective evolutionary algorithms for the deployment optimization strategy of a wsn in a smart building," *International Journal of Information Technology*, vol. 14, no. 1, pp. 333–344, 2022.
- [154] Batiactu, "The edge, vitrine lumineuse à amsterdam," <https://www.batiactu.com/edito/the-edge-vitrine-lumineuse-a-amsterdam-41605.php>, 2024, consulté le 3 novembre 2024.
- [155] C. Y. Wu, L. F. Wang, and C. H. Yang, "Random node deployment in wireless sensor networks : A review," *International Journal of Computer Applications*, vol. 48, no. 12, pp. 25–31, 2012.
- [156] M. K. Khedo and M. K. R. Nadarajah, "A survey of wireless sensor networks deployment methods," *Journal of Computer Networks and Communications*, vol. 2013, p. Article ID 949858, 2013.
- [157] P. M. Pradhan and S. K. Pal, "Multi-objective optimization of wireless sensor networks using genetic algorithm," *Journal of Networks*, vol. 10, no. 7, pp. 445–456, 2015.
- [158] F. Aurenhammer, "Voronoi diagrams : a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.
- [159] M. K. Gupta, R. M. Raghuwanshi, and K. S. V. T. K. Rao, "Using artificial potential fields in mobile wireless sensor networks," in *Proceedings of the 2nd International Conference on Advanced Computing, Networking and Security*, 2016, pp. 77–82.

- [160] Y. Shi and R. C. Eberhart, “A modified particle swarm optimizer,” in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1998, pp. 69–73.
- [161] Q. Wang, W. Wang, and L. Zhang, “Deployment strategies in wireless sensor networks : A survey,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 3, pp. 837–853, 2012.
- [162] R. J. Vanderbei, *Linear Programming : Foundations and Extensions*. Springer, 2001.
- [163] D. Bertsimas *et al.*, “Global optimization using branch-and-bound,” in *Handbook of Global Optimization*. Springer, 1995.
- [164] F. Aurenhammer, “Voronoi diagrams – a survey of a fundamental geometric data structure,” *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, Sep 1991.
- [165] S. Lindsey and C. Raghavendra, “Pegasis : Power-efficient gathering in sensor information systems,” in *IEEE Aerospace Conference Proceedings*, vol. 3, Mar 2002, pp. 1125–1130.
- [166] F. Glover, “Future paths for integer programming and links to artificial intelligence,” *Computers and Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.
- [167] X. Ma, X. Li, and J. Ma, “An improved pso-based approach for wireless sensor network deployment,” *Sensors*, vol. 18, no. 2, p. 518, 2018.
- [168] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [169] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. Zhang, “Multiobjective evolutionary algorithms : A survey of the state of the art,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.
- [170] T. M. M. de Almeida, G. F. M. Ferreira, and D. P. de Lima, “Multi-objective optimization in engineering design : A review,” *Journal of Manufacturing Science and Engineering*, vol. 136, no. 4, p. 041018, 2014.

- [171] Q. Zhang and H. Li, "Moea/d : A multi-objective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2009.
- [172] H. A. Taha, *Operations Research : An Introduction*, 10th ed. Upper Saddle River, NJ : Pearson, 2017.
- [173] A. Konnov and D. S. S. Teixeira, "Pareto optimality in multi-objective optimization," *Applied Mathematics and Computation*, vol. 183, no. 1, pp. 287–295, 2006.
- [174] K. Miettinen, "Nonlinear multiobjective optimization," *Springer Series in Operations Research*, 1999.
- [175] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms : Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [176] H. M. Hochman and J. D. Rodgers, "Pareto optimal redistribution," *The American Economic Review*, vol. 59, no. 4, pp. 542–557, 1969.
- [177] E. G. Talbi, "Metaheuristics : From design to implementation," in *Hybrid Metaheuristics*. Springer, 2002, pp. 1–15.
- [178] G. Berthiau and P. Siarry, "État de l'art des méthodes d'optimisation globale," *RAIRO-Operations Research*, vol. 35, no. 3, pp. 329–365, 2001. [Online]. Available : <https://doi.org/10.1051/ro:2001120>
- [179] J.-K. Hao, P. Galinier, and M. Habib, "Méthodes métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes," *Revue d'intelligence artificielle*, vol. 13, no. 2, pp. 283–324, 1999. [Online]. Available : <https://www.researchgate.net/publication/220080713>
- [180] Y. Cherruault, *Optimisation : méthodes locales et globales*. Presses Universitaires de France-PUF, 1999. [Online]. Available : <https://www.puf.com>
- [181] E. L. Lawler and D. E. Wood, "Branch-and-bound methods : A survey," *Operations Research*, vol. 14, no. 4, pp. 699–719, 1966. [Online]. Available : <https://doi.org/10.1287/opre.14.4.699>

- [182] M. Gendreau and J.-Y. Potvin, "Metaheuristics for the traveling salesman problem : A review," *Operations Research*, vol. 50, no. 4, pp. 609–628, 2002. [Online]. Available : <https://doi.org/10.1287/opre.50.4.609.12697>
- [183] E. Rouge and J. Puchinger, "Stochastic search and optimization : Methodologies and applications," *Journal of Heuristics*, vol. 5, no. 3, pp. 261–272, 1999. [Online]. Available : <https://doi.org/10.1023/A:1009688122595>
- [184] J. Pearl, *Heuristics : Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984. [Online]. Available : <https://www.pearson.com/us/higher-education/program/Pearl-Heuristics-Intelligent-Search-Strategies-for-Computer-Problem-Solving-2nd-Edition/PGM138719.html>
- [185] E.-G. Talbi, "Metaheuristics : From design to implementation," *John Wiley & Sons*, 2009. [Online]. Available : <https://www.wiley.com/en-us/Metaheuristics%3A+From+Design+to+Implementation-p-9780470741483>
- [186] K. Deb and S. Gupta, "Multi-objective optimization using evolutionary algorithms : A review," *Evolutionary Computation*, vol. 6, no. 2, pp. 183–217, 2001. [Online]. Available : <https://doi.org/10.1162/106365601750264476>
- [187] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.
- [188] I. Rechenberg, *Cybernetic Solution Path of an Experimental Problem*. Royal Aircraft Establishment Library Translation, 1965.
- [189] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor : University of Michigan Press, 1975.
- [190] C. Darwin, *On the Origin of Species*. London : John Murray, 1859, this seminal work introduces the concept of natural selection, heredity, and variation within species, which form the foundation of evolutionary biology.
- [191] J. J. Grefenstette, "Parallel adaptive algorithms for function optimization," Vanderbilt University, Computer Science Department, Tech. Rep. CS-81-19, 1981.

- [192] F. Alabsi and R. Naoum, “Fitness function for genetic algorithm used in intrusion detection system,” *International Journal of Applied Science and Technology*, vol. 2, no. 4, 2012.
- [193] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*. Springer, Berlin, 2003, vol. 53.
- [194] A. Baresel, H. Sthamer, and M. Schmidt, “Fitness function design to improve evolutionary structural testing,” in *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, 2002, pp. 1329–1336.
- [195] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.
- [196] K. Weicker and N. Weicker, “Basic principles for understanding evolutionary algorithms,” *Fundamenta Informaticae*, vol. 55, no. 3-4, pp. 387–403, 2003.
- [197] J. N. Richter, “On mutation and crossover in the theory of evolutionary algorithms,” Ph.D. dissertation, Montana State University-Bozeman, College of Engineering, 2010.
- [198] G. Olague, “Evolutionary computing,” in *Evolutionary Computer Vision*. Springer, Berlin, Heidelberg, 2016, pp. 69–140.
- [199] E.-G. Talbi, *Metaheuristics : from design to implementation*. John Wiley & Sons, 2009.
- [200] Unknown, “Etude des algorithmes génétiques et application aux données de protéomique,” Ph.D. dissertation, Université Montpellier I, 2007.
- [201] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm : NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [202] Q. Zhang and H. Li, “Moea/d : A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [203] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

- [204] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization : Formulation, discussion and generalization," in *Proceedings of the Fifth International Conference on Genetic Algorithms*, 1993, pp. 416–423.
- [205] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [206] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms : A comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [207] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2 : Improving the strength pareto evolutionary algorithm," in *Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems (EUROGEN 2001)*, 2001, pp. 95–100.
- [208] M. T. Emmerich and A. H. Deutz, "A tutorial on multiobjective optimization : fundamentals and evolutionary methods," *Natural Computing*, vol. 17, no. 3, pp. 585–609, 2018.
- [209] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu, "Interactive multiobjective optimization : A review of the state-of-the-art," *IEEE Access*, vol. 6, pp. 41 256–41 279, 2018.
- [210] I. Mampentzidou, E. Karapistoli, and A. A. Economides, "Basic guidelines for deploying wireless sensor networks in agriculture," in *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*. IEEE, 2012, pp. 864–869.
- [211] J. Brownlee, *Clever Algorithms : Nature-Inspired Programming Recipes*, 1st ed. Lulu.com, 2011.
- [212] O. Kramer, "Genetic algorithm essentials," in *Genetic algorithm essentials*. Springer, 2017, pp. 11–19.
- [213] S. Harizan and P. Kuila, "Design frameworks for wireless networks," in *Design frameworks for wireless networks*. Springer, 2020, pp. 257–280.
- [214] U. Mehboob, J. Qadir, S. Ali, and A. Vasilakos, "Soft computing," *Soft Computing*, vol. 20, no. 6, p. 2467, 2016.

- [215] F. Hoffmann, D. Medina, and A. Wolisz, "In 2017th international wireless communications and mobile computing conference." IEEE, 2011, pp. 1391–1396.
- [216] Y. Yoon and Y. Kim, "Ieee transactions on cybernetics," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, p. 1473, 2013.
- [217] M. Karatas, "Computer networks," *Computer Networks*, vol. 132, p. 129, 2018.
- [218] C. Liang and Y. Lin, "In 2018 ieee international conference on applied system invention (icasi)." IEEE, 2018, pp. 1272–1275.
- [219] M. Panhwar, Z. Deng, S. Khuhro, and D. Hakro, "Studies in informatics and control," *Studies in Informatics and Control*, vol. 27, no. 4, p. 461, 2018.
- [220] N. Hanh, H. Binh, N. Hoai, and M. Palaniswami, "Information sciences," *Information Sciences*, vol. 488, p. 58, 2019.
- [221] A. Tripathi, P. Gupta, A. Trivedi, and R. Kala, "In organizational efficiency through intelligent information technologies." IGI Global, 2013, pp. 125–144.
- [222] M. Rebai, H. Snoussi, F. Hnaïen, L. Khoukhi, and et al., "Computers operations research," *Computers Operations Research*, vol. 59, p. 11, 2015.
- [223] L. Dai and B. Wang, "In 2017 ieee 42nd conference on local computer networks (lcn)." IEEE, 2017, pp. 595–598.
- [224] A. Perez, "In 2018 32nd international conference on advanced information networking and applications workshops (waina)." IEEE, 2018, pp. 264–269.
- [225] S. Gupta, P. Kuila, and P. Jana, "Computers electrical engineering," *Computers Electrical Engineering*, vol. 56, p. 544, 2016.
- [226] M. Benatia, M. Sahnoun, D. Baudry, A. Louis, A. El-Hami, and B. Mazari, "Wireless personal communications," *Wireless Personal Communications*, vol. 94, no. 4, p. 2739, 2017.
- [227] S. Harizan and P. Kuila, "Wireless networks," *Wireless Networks*, vol. 25, no. 4, p. 1995, 2019.

- [228] H. ZainEldin, M. Badawy, M. Elhosseini, H. Arafat, and A. Abraham, “Journal of ambient intelligence and humanized computing,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–18, 2020.
- [229] P. Pal, R. Sharma, S. Tripathi, C. Kumar, and D. Ramesh, “Scientific reports,” *Scientific Reports*, vol. 11, no. 1, p. 1, 2021.
- [230] R. M. De Moraes, I. P. de Almeida, and L. R. Menezes, “A radio propagation model for dense wireless networks,” *International Journal of Wireless Information Networks*, vol. 26, no. 2, pp. 90–95, 2019.
- [231] P. F. Smith, S. Ganesh, and P. Liu, “A comparison of random forest regression and multiple linear regression for prediction in neuroscience,” *Journal of Neuroscience Methods*, vol. 220, no. 1, pp. 85–91, 2013.
- [232] D. Kong, Y. Chen, and N. Li, “Gaussian process regression for tool wear prediction,” *Mechanical Systems and Signal Processing*, vol. 104, pp. 556–574, 2018.
- [233] S. F. Taleb and N. Benalia, “Wireless sensor network deployment optimization for a smart farming application : comparison of two multi-objective evolutionary algorithms,” *ENP Engineering Science Journal*, vol. 3, no. 2, pp. 15–24, 2023.
- [234] A. K. Das, S. Sengupta, and S. Bhattacharyya, “A group incremental feature selection for classification using rough set theory based genetic algorithm,” *Applied Soft Computing*, vol. 65, pp. 400–411, 2018.
- [235] H. Mühlenbein and G. Paaß, “From recombination of genes to the estimation of distributions.” in *Proceedings of the 1st European Conference on Artificial Evolution*, 1991, pp. 178–187.
- [236] C. Blum and R. Roli, “Metaheuristics in combinatorial optimization : Overview and recent advances,” in *Handbook of Metaheuristics*, 2008, pp. 1–38.
- [237] J. J. Grefenstette, “Genetic algorithms for the optimization of control systems,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 4, pp. 422–430, 1985.

- [238] E. Mezura-Montes and C. A. Coello Coello, "Constraint-handling in nature-inspired numerical optimization : past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, 2011.
- [239] F. Glover and M. Laguna, *Tabu Search*. Kluwer Academic Publishers, 1998.
- [240] T. Stützle and H. H. Hoos, "Ant colony optimization for the traveling salesman problem," *Artificial Intelligence*, vol. 147, no. 1-2, pp. 5–26, 2001.
- [241] L. M. Gambardella, D. Taillard *et al.*, "Ant colony system for the tsp," *Journal of Operational Research*, vol. 87, no. 3, pp. 437–446, 1999.
- [242] E. McKendall and J. Shang, "Simulated annealing for the ant colony optimization algorithm," *European Journal of Operational Research*, vol. 173, no. 1, pp. 147–162, 2006.
- [243] R. Azimi, "An ant colony optimization algorithm for the examination timetabling problem," *Applied Soft Computing*, vol. 12, no. 8, pp. 2660–2670, 2012.
- [244] J. Suh and D. van Gucht, "A genetic algorithm with local search for the job shop scheduling problem," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*. IEEE, 2008, pp. 1238–1243.
- [245] C. Fleurent and J. Ferland, "Tabu search applied to graph partitioning," in *Proceedings of the 1994 IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 1994, pp. 104–108.
- [246] E.-G. Talbi *et al.*, "Hybrid evolutionary algorithms for the task scheduling problem," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 2. IEEE, 1999, pp. 2318–2325.
- [247] S. Lin, Y.-C. Kao *et al.*, "Simulated annealing for the quadratic assignment problem," *Computers Operations Research*, vol. 28, no. 12, pp. 1473–1494, 2001.
- [248] R. Tanese, "Distributed genetic algorithms," in *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann, 1989, pp. 434–439.

- [249] K. Ghédira and S. Hammami, “A hybrid approach for graph partitioning using tabu search and simulated annealing,” *International Journal of Computer Applications*, vol. 28, no. 7, pp. 25–30, 2011.
- [250] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [251] J. R. Koza, “Genetic programming : on the programming of computers by means of natural selection,” in *Proceedings of the 1st international conference on genetic algorithms and programming*. MIT Press, 1992, pp. 81–89.
- [252] I. Rechenberg, *Evolutionsstrategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, 1994.
- [253] K. Ghédira and S. Hammami, “Other works on co-evolutionary algorithms for various problems,” *International Journal of Computer Applications*, vol. 28, no. 7, pp. 30–35, 2011.
- [254] É. D. Taillard, “Parallel iterative search methods for vehicle routing problems,” *Networks*, vol. 23, no. 8, pp. 661–673, 1993.
- [255] É. D. Taillard and S. Voss, “A robust taboo search approach for the quadratic assignment problem,” *INFORMS Journal on Computing*, vol. 11, no. 4, pp. 285–291, 1999.
- [256] R. Krueger, “Genetic optimization algorithms and simulated annealing : a comparison and application to robust reliability problems,” in *Proceedings of the IEEE 2001 International Symposium on Semiconductor Manufacturing*. IEEE, 2001, pp. 381–384.
- [257] F. Abbattista, “A genetic algorithm to optimize the parameters of an ant colony system,” in *International Conference on Nature-Inspired Computation*. Springer, 2016, pp. 121–132.
- [258] K. Shahookar and P. Mazumder, “A genetic algorithm for vlsi physical design automation,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 5, pp. 500–511, 1990.

- [259] Y. Jin, “A comprehensive survey of fitness approximation in evolutionary computation,” *Soft computing*, vol. 9, no. 1, pp. 3–12, 2005.
- [260] M. S. Packiam, K. Deep, and S. Nagaraj, “Analytical model-based adaptive differential evolution for constrained optimization problems,” *Engineering Applications of Artificial Intelligence*, vol. 74, pp. 34–46, 2018.
- [261] T. Deb, K. M. Mallipeddi, and S. Dash, “Improving particle swarm optimization using the concept of stochastic diffusion search,” *Information Sciences*, vol. 179, no. 13, pp. 2207–2229, 2009.
- [262] S. Yang, X.-S. Yang, and R. Gandomi, “Bat algorithm : A novel approach for global engineering optimization,” *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.
- [263] S. F. Taleb, N. E.-H. Benalia, and R. Sadoun, “Fitness approximation with rf algorithm dedicated to wsn node deployment for a soil monitoring application,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 43, no. 2, pp. 72–86, 2023.
- [264] M. S. Packiam, K. Deep, and S. Nagaraj, “Analytical model-based adaptive differential evolution for constrained optimization problems,” *Engineering Applications of Artificial Intelligence*, vol. 74, pp. 34–46, 2018.
- [265] Y. Jin, “Model management for evolutionary computation with approximate fitness models,” *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 4, pp. 385–397, 2005.
- [266] Y. Zhang, L. Wang, Y. Ruan, and R. Zhang, “Deep learning for health informatics,” *Health Information Science and Systems*, vol. 7, no. 1, pp. 1–10, 2019.
- [267] N. Altman and M. Krzywinski, “An introduction to kernel and nearest-neighbor non-parametric regression,” *The American Statistician*, vol. 46, no. 2, pp. 175–182, 1992.
- [268] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth International Group, 1986.

- [269] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory*. Springer, 1997, pp. 23–37.
- [270] W. YOU, A. Saidi, A. Zine, and M. Ichchou, "Mechanical reliability assessment by ensemble learning," *Vehicles*, vol. 2, pp. 126–141, 02 2020.
- [271] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [272] L. Breiman and A. Cutler, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [273] C. Ulmer, "Organization techniques in wireless in-situ sensor networks," Georgia Tech Research Project, Tech. Rep., 2003, rapport technique. [Online]. Available : <https://dl.acm.org/doi/abs/10.5555/1515915.1515947>
- [274] Y. Jin and B. Sendhoff, "Reducing fitness evaluations using clustering techniques and neural network ensembles," in *Genetic and Evolutionary Computation Conference*. Springer, 2004, pp. 688–699.
- [275] P. K. Nain and K. Deb, "A computationally effective multi-objective search and optimization technique using coarse-to-fine grain modeling," in *IN 2002 PPSN WORKSHOP ON EVOLUTIONARY MULTIOBJECTIVE OPTIMIZATION A COMPREHENSIVE SURVEY OF FITNESS APPROXIMATION IN EVOLUTIONARY COMPUTATION 13 TION*. Citeseer, 2002.
- [276] D. Buche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with gaussian process fitness function models," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 2, pp. 183–194, 2005.
- [277] D.-J. Wang, F. Liu, Y.-Z. Wang, and Y. Jin, "A knowledge-based evolutionary proactive scheduling approach in the presence of machine breakdown and deterioration effect," *Knowledge-Based Systems*, vol. 90, pp. 70–80, 2015.

- [278] J.-W. Yoon and S.-B. Cho, "Fitness approximation for genetic algorithm using combination of approximation model and fuzzy clustering technique," in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–6.
- [279] Y. Liang, Z. Ren, Y. Yang, A. Chen, D. Guo, and B. Pang, "A mahalanobis distance-based fitness approximation method for estimation of distribution algorithms in solving expensive optimization problems," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 1608–1613.
- [280] M. A. Benatia, M. Sahnoun, A. Louis, and D. Baudry, "Hybrid meta-heuristics for the optimization of the deployment of wireless sensor networks (wsn)," EasyChair, Tech. Rep., 2019.
- [281] Q. Zhang, H. Li, and K. Deb, "Multi-objective optimization problems with complicated pareto sets, moea/d and nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 921–932, 2010.
- [282] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.
- [283] C. A. C. Coello and M. Lechuga, "Evolutionary algorithms for solving multi-objective problems," *Evolutionary Computation*, vol. 12, no. 3, pp. 283–319, 2006.
- [284] J. Bader and E. Zitzler, "Heterogeneous parallel evolutionary algorithms for multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 482–490, 2005.
- [285] Z. Michalewicz and D. B. Fogel, "How to solve it : Modern heuristics," *Springer*, pp. 342–350, 1996.
- [286] E. Cantu-Paz, "Migration policies for parallel genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 214–228, 2000.
- [287] E. Alba and J. R. Dorronsoro, "Parallelism in genetic algorithms : A survey," *International Journal of Computational Intelligence and Applications*, vol. 1, no. 4, pp. 511–528, 1999.

- [288] K. Bethke, "Parallel genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, no. 4, pp. 411–417, 1976.
- [289] T. Forgarty and W. R. MacGregor, "The genetic algorithm as a parallel method," in *Proceedings of the Third International Conference on Genetic Algorithms*, 1991, pp. 60–67.
- [290] D. Abramson, "Parallel genetic algorithms : A review," in *Proceedings of the Second International Conference on Genetic Algorithms*, 1992, pp. 12–19.
- [291] S. Hauser and W. R. MacGregor, "A parallel genetic algorithm for traveling salesman problem," in *Proceedings of the Fourth International Conference on Genetic Algorithms*, 1994, pp. 60–67.
- [292] J. Bossert, "On the existence of stable populations in a genetic algorithm," *Journal of Theoretical Biology*, vol. 14, no. 1, pp. 1–10, 1967.
- [293] L. Grosso, "Multiple population genetic algorithms," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1985, pp. 59–66.
- [294] C. Braun, "Parallel genetic algorithms with multiple populations," in *Proceedings of the International Conference on Genetic Algorithms*, 1990, pp. 115–120.
- [295] C. Pettey, "Diffusion genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 1997, pp. 257–263.
- [296] —, "Massively parallel genetic algorithms," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1992, pp. 35–40.
- [297] M. J. Flynn, "Some computer organizations and their effectiveness," *IEEE Transactions on Computers*, vol. 21, no. 9, pp. 948–960, 1972.
- [298] M. J. Quinn, *Parallel Computing : Theory and Practice*. McGraw-Hill, 2011.
- [299] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design : The Hardware/Software Interface*, 5th ed. Morgan Kaufmann, 2017.
- [300] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts*, 9th ed. John Wiley & Sons, 2011.

- [301] R. Battiti and D. Brunato, "Introduction to neural computation," *Neural Computation*, vol. 12, no. 6, pp. 1415–1432, 2000.
- [302] O. Maitre, "Gpgpu for evolutionary algorithms," Ph.D. dissertation, Strasbourg, 2011.
- [303] J. E. Savage, *Models of Computation—Exploring the Power of Computing*, 1998.
- [304] R. W. Hockney and C. R. Jesshope, *Parallel Computers 2 : Architecture, Programming and Algorithms*. CRC Press, 2019.
- [305] E. Hermann, "Interactive physical simulation on multi-core," in *Proceedings of the ACM SIGGRAPH Conference*, 2010.
- [306] E. Alba, G. Luque, and S. Nesmachnow, "Parallel metaheuristics : recent advances and new trends," *International Transactions in Operational Research*, vol. 20, no. 1, pp. 1–48, 2013.
- [307] L. Snyder, *GPU Computing*. Morgan Kaufmann, 2006.
- [308] S. Chikkagoudar, K. Wang, and M. Li, "Genie : A software package for gene-gene interaction analysis in genetic association studies using multiple gpu or cpu cores," *BMC research notes*, vol. 4, p. 158, 05 2011.
- [309] NVIDIA, "Cuda c programming guide," 2012. [Online]. Available : <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>
- [310] O. Maitre, N. Lachiche, P. Clauss, L. Baumes, A. Corma, and P. Collet, "Efficient parallel implementation of evolutionary algorithms on gpgpu cards," in *European Conference on Parallel Processing*. Springer, Berlin, Heidelberg, 2009, pp. 974–985.