REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : RTIC16/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Réseaux et Technologies de l'Information et de la Communication(RTIC)**

Résoudre le problème de déploiement 2D des réseaux de capteurs par l'algorithme NSGA-2

Par :

BEZZIOU SARRA

Soutenu le 27/06/2022 devant le jury composé de :

KELFALI	Toufik	M.A.A	Président
ABABSA	Tarek	M.C.B	Rapporteur
ZOUAI	Meftah	M.A.B	Examinateur

Année universitaire 2021-2022

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة محمد خيضر بسكرة

تصريح شرفي

(خاص بالالتزام بقواعد النزاهة العلمية لإنجاز بحث)

أنا الممضي أسفله، السيد (ة): **بـــزبـــو ســـارة** الصفة: .**طالبة**

الحامل لبطاقة التعريف الوطنية رقم: **119790232011850006**

والصادرة بتاريخ: **2017/01/16**

المسجل بكلية: العلوم الدقيقة و علوم الطبيعة و الحياة قسم: الإعـــــــلام الآلـي

والمكلف بإنجاز مذكرة تخرج في الماستر عنوانها:

Résoudre le problème de déploiement 2D des réseaux de capteurs par l'algorithme NSGA-2

أصرح بشرفي أني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات المهنية والنزاهة الأكاديمية المطلوبة في إنجاز البحث المذكور أعلاه.

التاريخ: 2002/06/23

توقيع المعني:

ملخص

خلال مشروع نهاية الدراسة هذا ، تناولنا مشكلة نشر شبكات الاستشعار اللاسلكية في مكان محدد .ان هدفنا هو تطوير أداة من شأنها إنشاء مخطط نشر فعال ، وتقديم خدمة ذات جودة عالية بدأنا بتقديم شبكات الاستشعار اللاسلكية .ثم قمنا بنمذجة المعلمات الرئيسية لنشر شبكات الاستشعار اللاسلكية ، وهي التكلفة والاتصال والتغطية والتغطية الزائدة .بعد ذلك ، اخترنا خوارزمية تطورية متعددة الأهداف قمنا بتنفيذها من أجل حل مشكلة النشر .حصلنا على نتائج مرضية من المحاكاة
الكلمات الرئيسية : شبكة الاستشعار اللاسلكية ، النشر ، الاتصال ، التغطية ، التحسين متعدد الأهداف، الخوارزمية التطورية ، الخوارزمية الجينية.

Abstract

During this graduation project, we dealt with the problem of deploying wireless ensor networks. Our goal was to develop a tool that would stablish an efficient deployment scheme, offering good quality of service. We started with the introduction of wireless sensor networks. Then, we modeled the key parameters of a wireless sensor network deployment, namely cost, connectivity, coverage and over-coverage. Thereafter, we selected an evolutionary multi-objective algorithm that we implemented in order to solve the deployment problem. We obtained satisfactory results from the simulations.

Keywords: wireless sensor network, deployment, connectivity, coverage, multi-objective optimization, evolutionary algorithm, genetic algorithm.

Résumé

Nous avons traité au cours de ce projet de fin d'étude le problème de déploiement des réseaux de capteurs sans fil. Notre objectif était de développer un outil qui permette d'établir un schéma de déploiement efficace, offrant une bonne qualité de service.

Nous avons commencé par l'introduction des réseaux de capteurs sans fil. Puis nous avons modélisé les paramètres clés d'un déploiement d'un réseau de capteurs sans fil, à savoir le coût, la connectivité, la couverture et la sur-couverture. Par la suite, nous avons sélectionné un algorithmes évolutionnaires multi-objectifs NSGA II que nous avons implémenté afin de résoudre le problème de déploiement. Nous avons obtenu à l'issu des simulations des résultats satisfaisants.

Mots clés: réseau de capteurs sans fil, déploiement, connectivité, couverture, optimisation multi-objectif, algorithme évolutionnaire, algorithme génétique.

Remerciement

Avant tout, nous remercie Allah de nous avoir donné la force, la volonté et le courage pour l'accomplissement de notre tâche et nous accorder le courage, la patience pour arriver à cette fin.

C'est ainsi qu'à travers ce mémoire, nous exprimons toute notre gratitude et remerciements à notre encadreur ***Mr. Ababsa Tarek*** *pour sa disponibilité, son aide, ses conseils.*

Toutes nos reconnaissance set remerciements distingués s'adressent à ***Mr. Tebermassine Okba*** *chef du département informatique, à toute son équipe administrative et particulièrement* ***Mme Taa allah latifa*** *ainsi qu'à tous nos enseignants de la faculté des sciences et sciences exactes de l'Université Mohamed Khider de Biskra en particulièrement* ***Mme .Brima Salima*** *Pour ses conseils et son aide.*

Je désire aussi remercie spécialement à l'enseignant ***Mme Nour El-Hoda benalia*** *pour sa disponibilité, son aide et ses conseils.*

On n'oublie pas ***mon marie*** *pour leur contribution, leur soutien et leur patience, et a toute personne ayant contribué de pré ou de loin à la réalisation de ce mémoire, et nous n'oublierons aussi jamais tous nos camarades de la promotion 2022 LMD du département informatique.*

Table de matières

Liste des tableaux

Liste des figures

Liste des équations

Liste des tableaux

Liste des équations

Liste des figures

Introduction générale

Les capteurs existent depuis plusieurs années dans les domaines de l'industrie tels que l'aéronautique, l'automobile. On observe actuellement une forte recrudescence de ce type d'équipements qui sont interconnectés pour former des réseaux de capteurs. Avant, ils étaient reliés directement à leur base de traitement par une liaison filaire, ces capteurs sont maintenant de plus en plus interconnectés par ondes radios (ZigBee par exemple). En 2003, selon le magazine Technology Review du MIT, le réseau de capteurs sans fil est l'une des dix nouvelles technologies qui bouleverseront le monde et notre manière de travailler et de vivre. Chaque capteur collabore avec ses voisins via des liens sans fil, et si nécessaire doit retransmettre les informations venant de capteurs trop éloignés pour communiquer directement avec la station de base qui à son tour retransmet ces données, après les avoir traité, au utilisateur final du réseau

Notre problématique s'inscrit dans ce contexte. Elle vise à étudier la conception et le déploiement de ce type de réseaux. En effet, pour pouvoir déployer un RCSF, il faut respecter un certain nombre de contraintes dans le but d'aboutir à un réseau fonctionnel qui réponde aux besoins de l'application à laquelle il est destiné. Généralement, ces contraintes sont attachées directement à la connectivité de l'ensemble des capteurs, à la couverture de la zone etc. Le déploiement d'un RCSF consiste à déterminer le nombre et les positions des nœuds. Ces derniers doivent former un réseau respectant les taux de connectivité, couverture et sur-couverture désirés.

Avec l'avancée technologique et le développement des ordinateurs qui sont devenus très performants et dotés d'une plus grande capacité de mémoire et de calcul, plusieurs outils de déploiement ont été proposé pour aider les concepteurs à trouver la topologie optimale d'un réseau en utilisant plusieurs notions et techniques. Ces dernières vont être abordées dans les chapitres de notre document Ces dernières décennies, les méthodes d'optimisation et les méta-heuristiques ont été utilises pour concevoir et déployer les RCSFs. Dans ce sens et à part quelques travaux, les solutions proposées traitent que d'une seule ou deux contraintes au maximum.

Alors que, le développement d'un outil capable de traiter trois objectifs ou plus est une nécessité primordiale pour la conception d'un RCSF efficace. Pour cela, une bonne modélisation du système et un bon choix de la méthode d'optimisation de déploiement doivent être fait.

Le but de notre travail est de développer une application qui permette d'établir un plan de déploiement d'un RCSF dans un espace. Pour ce faire, il est nécessaire en premier lieu de modéliser les différents critères de déploiement. Ces critères sont : le coût de déploiement qui est proportionnel au nombre de nœuds capteurs et qui doit être minimisé, la couverture qui doit être complète pour une surveillance sans faille, la connectivité du réseau, qui elle aussi doit être complète pour assurer une bonne qualité de service, et finalement, la sur-couverture pour éviter de déployer des nœuds sans intérêt. Ensuite, une méthode d'optimisation multi-objectif doit être appliquée afin de déterminer le nombre et l'emplacement des différents nœuds constituants le réseau.

Pour réaliser notre objectif, nous allons organiser ce mémoire en deux parties :

- ✓ La première partie composée de deux chapitres :
  - Le premier chapitre sous le titre « les réseau de capteur sans fil et le déploiement » on s'est intéressé à une étude sur les réseaux de capteur sans fil et la modélisation des différents critères de déploiement.
  - Le deuxième chapitre sous le titre « les algorithmes génétique » nous présentons les concepts de base des algorithmes génétiques qui sont au cœur de ce travail et la méthode d'optimisation utilisées pour la résolution du problème de déploiement des RCSFs multi-objectifs NSGA II.
- ✓ La deuxième partie contient deux chapitres vont être consacrés a la conception du système ou nous essayerons de fonder la composition de notre système, et l'implémentation ou nous allons construire notre modèle et montre nos résultats.

Nous achevons notre travail par une conclusion générale et quelques perspectives.

Chapitre I

1. Les réseaux de capteurs sans fils et le déploiement

1.1 Introduction

Depuis leur création, les réseaux de capteur sans fil ont connu un succès sans cesse croissant au sein des communautés scientifiques et industrielles. Grâce à ses diverses avantages, cette technologie a pu s'instaurer comme acteur incontournable dans les architectures réseaux actuelles Le but général d'un RCFS est la collecte d'un ensemble de paramètres de l'environnement, telles que la température ou la pression de l'atmosphère, afin de les acheminer vers des points de traitement.

Les RCSFs sont utilisés dans plusieurs domaines et pour diverses applications. Les capacités limitées des capteurs en termes de stockage d'énergie, traitement des données, et coût des communications font que les RCSFs sont sujets à de nombreux défis comme le déploiement des capteurs.

Dans ce chapitre, nous commencerons par définir les réseaux des capteurs sans fils en général après, décrire les Topologie des réseaux et Standards de communication des RCFSs utilisés. Ensuite, nous présenterons leurs défis posés tel que le Déploiement d'un RCSF.

1.2 Architecture d'un réseau de capteurs sans filles

1.2.1 Nœuds d'un RCSF

Un RCSF est compose d'un grand nombre de nœuds déployés dans une région d'intérêt, ajoutes à ceux-là, des nœuds relais ou sink ainsi qu'une ou plusieurs stations de base [24][28].

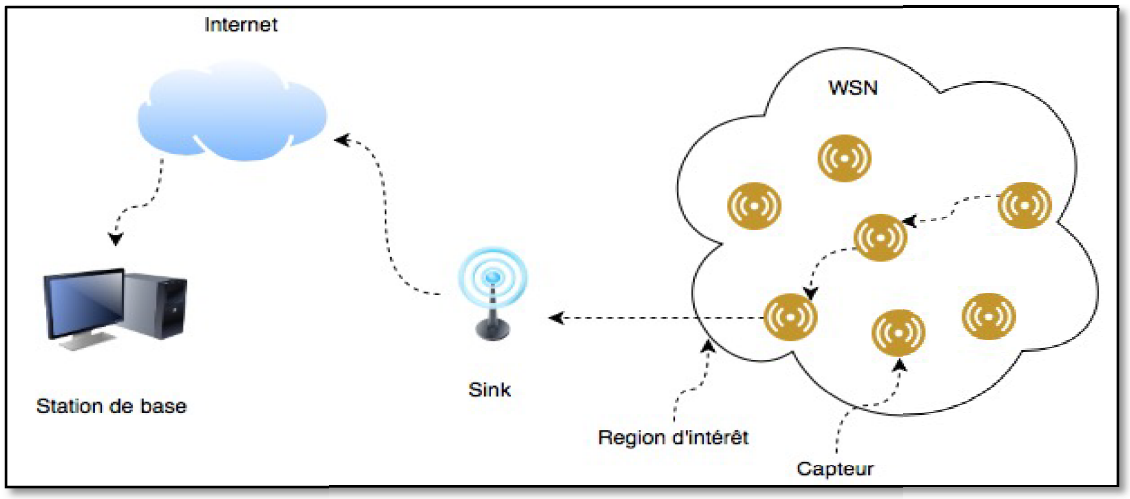


Figure 1.1 : Architecture d'un réseau de capteurs sans fil

- **Station de base** : c'est un point centralise de contrôle dans le réseau. Elle est employée pour extraire des informations du réseau et envoyer des informations de contrôle au différent nœud du réseau.
- **Sink** : aussi appelé nœud puits ou routeur, le sink sert de passerelle entre le champ de captage et la station de base. Ce dispositif est également un capteur mais qui peut 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.
- **Nœud capteur** : les nœuds capteurs recueillent les informations liées à leur environnement de déploiement. Il existe deux types de nœud capteur : les capteurs axes et les capteurs mobiles. Un nœud capteur est généralement composé des cinq modules suivants :
  - Module d'acquisition des données ;
  - Module de traitement ;
  - Module de stockage ;
  - Module de communication ;
  - Module d'énergie.

1.2.2 Topologie des réseaux

Un RCSF peut être organisé selon différents types de topologie dont voici les principales. La figure (1.2) illustre ces différentes topologies.

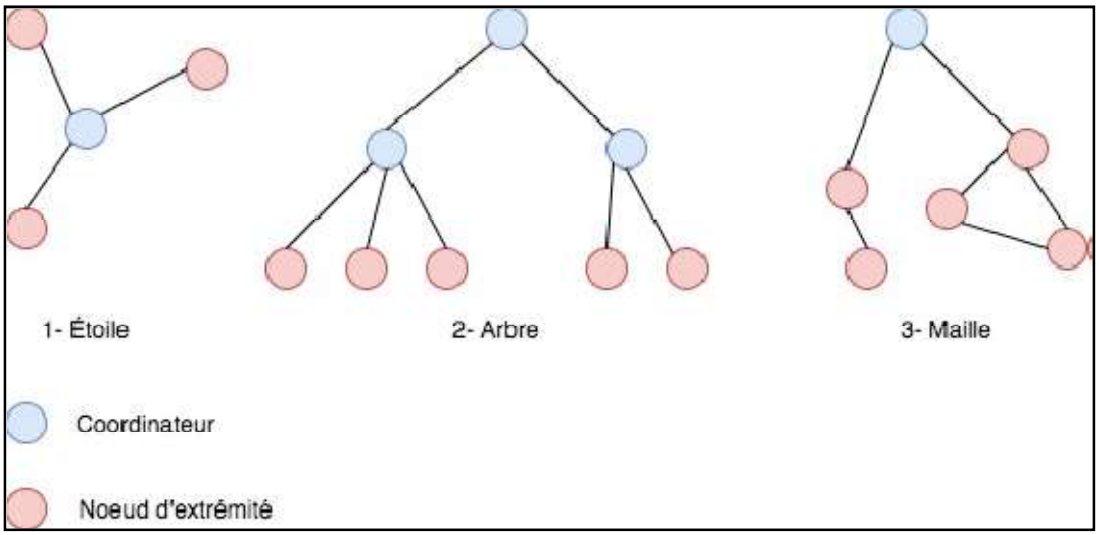


Figure 1.2 : Topologies des réseaux

1.2.2.1 Etoile

Le réseau est constitué d'un ensemble de nœuds directement relies à un coordinateur central. Les nœuds ne sont pas autorises à interagir entre eux et communiquent via le coordinateur central [2]. 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 [3]. L'inconvénient de cette topologie est que si le coordinateur central est défectueux, tout le réseau tombe en panne.

1.2.2.2 Arbre

Le réseau contient un nœud racine, de nombreux routeurs et des périphériques d'extrémité. Tous les nœuds sont lies sous forme d'arborescence. Les nœuds d'extrémité sont directement lies au coordinateur et aux routeurs en tant que noud enfants. Un terminal ne peut interagir avec un autre terminal que via son nœud parent. L'inconvénient de la topologie arborescente et que si l'un des parents devient désactivé, les enfants de celui-ci ne peuvent pas interagir avec d'autres périphériques du réseau [2].

1.2.2.3 Maille

Une topologie maillée est autogérée, c'est-à-dire que pendant la transmission, si l'un des chemins échoue, le nœud découvrira un chemin alternatif vers le nœud de destination. Tout appareil source peut interagir avec n'importe quel appareil de destination du réseau [2]. L'inconvénient de cette topologie et que l'augmentation du

nombre de sauts d'une communication à 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 [4].

1.2.2.4 Topologie hybride

La topologie hybride [43] 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.

1.3 Standards de communication

1.3.1 ZIGBEE

Zigbee est un standard utilisé dans les communications à très faible puissance et sur des distances réduites notamment dans les réseaux de capteurs sans fils. Il prévoit des vitesses allant jusqu'à 250 kbps sur une plage de 10 à 100 m et utilise la bande de fréquence 2,4 GHz. Il repose sur les couches basses du standard IEEE 802.15.4. Le protocole Zigbee peut effectuer les opérations suivantes : Découverte du voisinage, création de la topologie, adressage et routage [22].

1.3.2 Wi-Fi

Wi-Fi est un ensemble de protocoles de communication sans fil régis par les normes IEEE 802.11. Les normes Wi-Fi, permettent de créer des réseaux locaux sans fils à haut débit. Pour les réseaux de capteurs sans fils une nouvelle version de Wi-Fi appelée Wi-Fi à faible puissance "Wi-Fi low power" est utilisée. Elle est caractérisée par une faible consommation d'énergie [7].

1.3.3 UWB

Ce protocole peut procurer un débit de 55 Mbps pour une distance allant jusqu' à 100 mètres. Il utilise la bande de 2,4 GHz ce qui élimine le risque d'interférences avec les autres types de réseaux. La norme IEEE 802.15.3 est caractérisée par une bonne qualité de service étant donne qu'elle inclut le protocole TDMA (Time Division Multiple Access). Le standard UWB bénéficie de très bonnes performances en termes de sécurité [8].

1.3.4 Bluetooth

Bluetooth utilise des ondes radios UHF destinées à simplifier les connexions entre les appareils électroniques. Cette norme a été proposée pour transmettre la voix et les données. Elle est peu utilisée dans les réseaux de capteurs sans fils parce que très gourmande en énergie et dispose d'une topologie réseau complexe [9].

1.3.5 Wibree

Wibree, plus connu sous Bluetooth Low Energy (BLE), est une technique de transmission sans fil crée par Nokia. Ce standard est base sur Bluetooth. Wibree consomme 10 fois moins d'énergie que Bleutooth pour un même débit. Cela rend possible l'utilisation de cette technique dans des équipements à faible puissance comme les capteurs sans fils. Sa limite principale est la faible portée de communication : 5-10m [10].

1.4 Domaines d'application des RCSFs

Les RCSFs ont connu un grand succès dans plusieurs domaines voir figure (1.3) et ce grâce au rétrécissement de la taille des capteurs, leur facilité d'adaptation à n'importe quel milieu et enfin la communication sans fil.

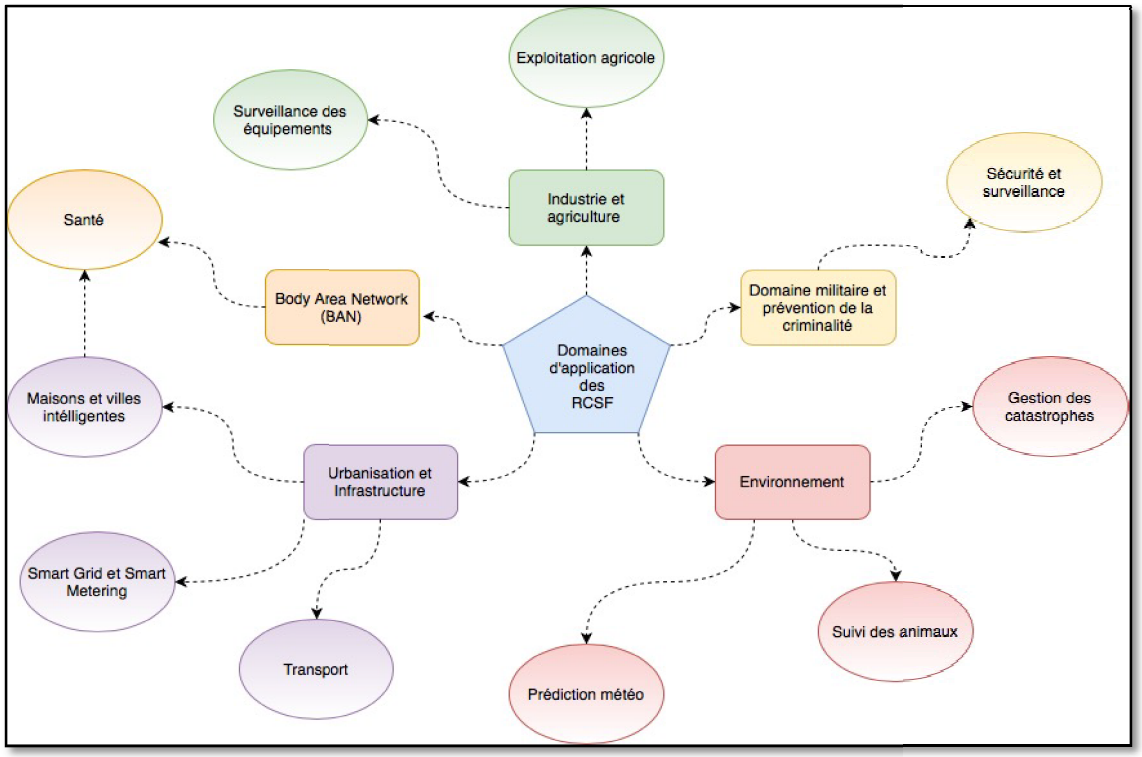


Figure 1.3 : Taxonomie des applications des RCSFs.

1.5 Mesure de performance des RCSFs

De nombreuses mesures de performances doivent être prises en compte dans les RCSFs. Les facteurs qui ont une influence sur la qualité de service (QoS) d'un RCSF sont la consommation d'énergie, l'évolutivité, la fiabilité, la puissance du signal, la précision, la latence, la couverture, et la connectivité. Cette liste est bien évidemment non exhaustive. Ci-dessous une brève discussion de chaque métrique [11] :

1.5.1 Consommation d'énergie :

La consommation d'énergie est un facteur important à prendre en compte lors de l'implémentation d'algorithmes dans les WSNs. Diminuer la consommation d'énergie par nœud peut être obtenue en réduisant le nombre d'échanger de messages entre les nœuds. En outre, planifier des intervalles de sommeil pour les nœuds redondants, tout en laissant les nœuds restants actifs pour maintenir la couverture du réseau et la connectivité, augmente la durée de vie du réseau. Outre la diminution de la taille des messages transmis entre les nœuds, la sélection de la meilleure méthode de routage et la réduction de la mobilité des nœuds réduisent la consommation d'énergie dans les RCSFs.

1.5.2 Evolutivité et Fiabilité

L'évolutivité est la capacité du réseau à être étendue par ajout de nœuds tout en maintenant les performances du réseau. La fiabilité représente la livraison des données. L'évolutivité et la fiabilité sont des problèmes critiques dans les RCSFs en raison du très haut nombre de nœuds.

1.5.3 Puissance du signal

La puissance du signal est une mesure de la qualité de la liaison. On utilise la distance entre deux nœuds pour déterminer l'accessibilité des nœuds pendant le processus de communication. L'indication de la puissance du signal reçu (RSSI) décrit la force d'un signal sans fil et est définie comme suit :

$$\boldsymbol{RSSI = -10 * n * log10(d) + p} \qquad 1.1$$

Ou *d* est la distance en mètre, *n* est la constante de propagation et *p* est la puissance en mode réception (dBm).

1.5.4 Précision et latence :

La latence est une mesure du retard. Elle mesure le temps qu'il faut pour une donnée pour arriver à la destination sur le réseau. La précision représente l'efficacité des données livrées à la destination. La réduction du retard grâce à la transmission des données garantit la précision du réseau.

1.5.5 Couverture et connectivité

La couverture est l'une des mesures du QoS (Quality of service) des RCSFs. La couverture sans connectivité complète diminue la qualité du RCSF car, une défaillance au niveau de la connectivite influe directement sur la réception des données par le nœud récepteur. De plus, la connectivité sans couverture provoque des points non couverts dans la zone cible et des trous de couverture apparaissent. Par conséquent, la couverture et la connectivité doivent être prises en compte simultanément lors du déploiement d'un RCSF.[28]

1.6 Défis posés par les RCSFs

Les RCSFs sont utilisés dans plusieurs domaines (voire la figure(1.3)) et pour diverses applications. Les capacités limitées des capteurs en termes de stockage d'énergie, traitement des données, et coût des communications font que les RCSFs sont sujets à de nombreux défis. Beaucoup de recherches ont été conduite dans le but de pallier à ces limites. Notamment en ce qui concerne la localisation des capteurs [12], le stockage des données, le routage des données des capteurs vers le sink, la consommation d'énergie, la fiabilité des réseaux ainsi que leur évolutivité, les interférences et le déploiement des capteurs [11].

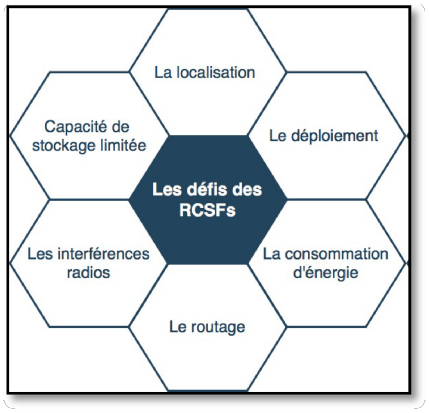


Figure 1.4 : Défis des réseaux de capteurs sans fils.

1.7 Déploiement d'un RCSF

Le déploiement des capteurs consiste à placer des nœuds capteurs dans la région d'intérêt de manière à avoir une couverture totale de celle-ci, tout en assurant une connectivité avec le ou les nœuds relais (sinks). Elle est considérée comme l'un des problèmes les plus critiques dans la mise en œuvre des RCSFs. La couverture est parmi les problèmes les plus importants pour la réalisation d'un RCSF. Elle est affectée essentiellement par le rayon de détection (Rd) du nœud capteur. D'autre part, la connectivité dépend de la portée de communication (rayon de communication : Rc) des nœuds capteurs et de leur disposition dans l'espace. La consommation d'énergie est un facteur important qui doit être pris en compte. La planification des modes actif et veille des nœuds de capteurs après le déploiement optimal des nœuds minimise la consommation d'énergie et prolonge la durée de vie du réseau.[28]

1.8 Critères de déploiement

1.8.1 Espace de déploiement

Les capteurs sans fils peuvent être placés dans différents endroits (murs, plafond, meubles . . .) . L'espace de déploiement peut donc être modélisme en 3D ou bien en 2D. Nous considérons dans notre travail un espace de déploiement en 2D.

L est la longueur de la zone et l sa largeur.

1.8.2 Coût de déploiement

L'objectif principal de notre travail est de réduire au maximum le coût du RCSF à déployer tout en maximisant sa qualité de service. Il est donc très important de modéliser le coût de déploiement afin de l'intégrer dans notre fonction objectif. Le coût de déploiement comprend le coût d'achat des capteurs et leurs coûts d'installation. Si on considère un RCSF composé de l'ensemble des nœuds S = s_1,s_2, s_3,….. s_m capteurs. Le coût de déploiement peut être calculé comme suit :

$$\text{Coût} = \text{Cs} . ||S|| \qquad 1.2$$

Cs : Coût d'achat et d'installation d'un nœud capteur ;

||S||: Le nombre de nœuds capteurs déployés ;

Matrice de déploiement comme suit :

$$D(i,j) = \begin{cases} 1 & \text{si un capteur ou un relais est déployé dans la cellule ;} \\ 0 & \text{sinon.} \end{cases} \quad 1.3$$

La fonction coût peut être obtenue à partir de la matrice de déploiement comme suit

$$Coût = \sum_{i=0}^{l-1} \sum_{j=0}^{L-1} D(i,j) \quad 1.4$$

Il s'agit d'un coût unitaire qui sera ensuite multiplié par le coût des capteurs pour obtenir le coût de déploiement des capteurs.

1.8.3 Couverture

On dit qu'une région ou qu'un point est couvert par un capteur si cette région ou ce point se trouve dans la zone de couverture d'un ou plusieurs capteurs actifs [13], il existe trois types de couverture :

1.8.3.1 Couverture de cham

L'objectif de ce type de couverture est que chaque point de la région d'intérêt soit dans le champ de couverture d'au moins un capteur. Comme on peut le voir dans la figure (1.5), la région d'intérêt (le carré) est complètement couverte par les capteurs déployés. En fonction de l'application, il existe deux types de couverture de champ : couverture complète et couverture partielle.

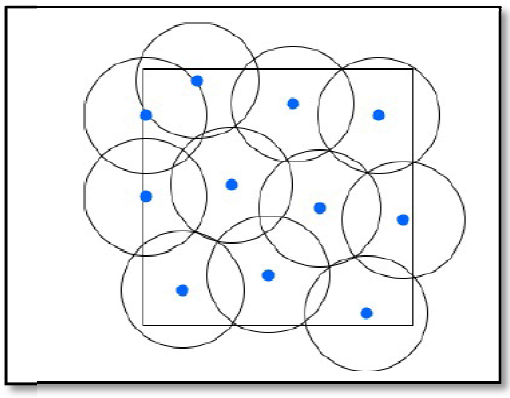


Figure 1.5 : Couverture complète de la région d'intérêt.

Dans le cas d'une couverture partielle, les capteurs sont déployés de manière à couvrir un pourcentage p de la surface totale. Adopter une couverture partielle permet de réduire le coût de déploiement et d'augmenter la durée de vie du réseau de capteurs

[13]. La couverture partielle est utilisée dans des applications liées à l'environnement, tel que le relevé de température dans une région donnée et la détection des feux de forêts.

1.8.3.2 Couverture de cible (Point d'intérêt)

Lorsqu'il s'agit de surveiller des points spécifiques d'une région, on a recours à une couverture de cible. Comme on peut le voir dans la figure (1.6), les cibles représentées par des hexagones noirs se trouvent dans le rayon de couverture d'au moins un capteur. Le nombre de cibles étant fixe, le coût de déploiement du réseau se voit réduit puisqu'il s'agit d'une couverture partielle qui ne nécessite pas un nombre élevé de capteurs.

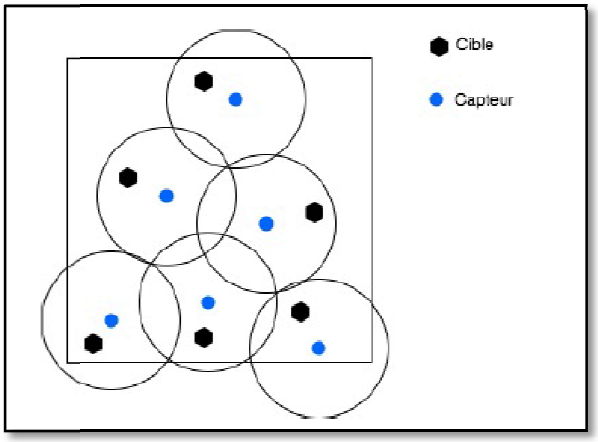


Figure1.6 : Couverture des points d'intérêts (cibles).

1.8.3.3 Couverture de barrière

Dans un schéma de couverture de type barrière les nœuds sont déployées sous forme de barrière sur un chemin spécifique. Ces nœuds signalent d'éventuelles activités suspectes, la couverture de barrière convient à la détection d'intrusions, aux applications militaires et à la surveillance des frontières. En fonction des applications on distingue deux types de régions d'intérêts : ROI en forme de ceinture ouverte, ROI en forme de ceinture fermée [55]. L'objectif de ce type de couverture est de minimiser la probabilité d'intrusion à travers la barrière.

1.8.4 La sur-couverture

Afin de réduire le coût de déploiement du réseau de capteurs sans fil et de diminuer les interférences entre les capteurs, il est nécessaire de réduire la surface des

zones dites sur-couvertes, c'est à dire les zones couvertes par deux ou plusieurs capteurs comme on peut le voir dans la figure. Pour ce faire, nous allons utilise une des représentations existantes de la sur-ouverture, définit comme suit :

$$Sur{-}couv(i,j) = \begin{cases} 1 & \text{si la cellule (i,j) est couverte par au moins deux capteurs ;} \\ 0 & \text{sinon.} \end{cases} \quad 1.5$$

Nous implémentons par la suite une fonction qui calcule le taux de sur-couverture à l'aide de la formule suivante :

$$Sur - couv_{RI} = \frac{\sum_{i=0}^{l-1}\sum_{j=0}^{L-1} Sur - couv(i,j)}{L \cdot l} \cdot 100$$

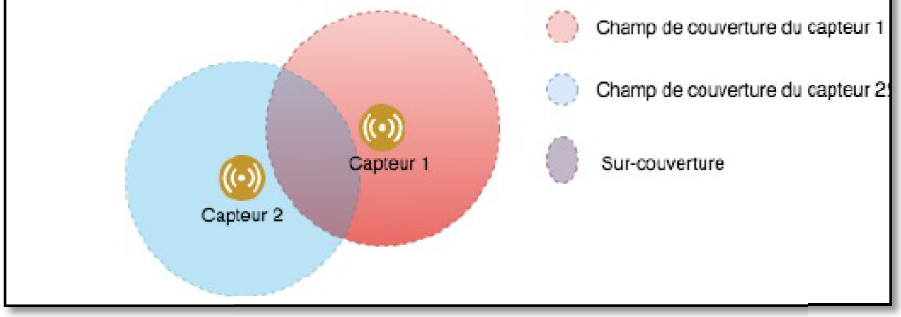


Figure 1.7 : Modélisation de la zone de sur couverture

1.8.5 Détection des événements

La détection des événements et la collecte des données sont parmi les principaux objectifs des réseaux de capteurs sans fils. Plus on s'éloigne du capteur plus sa capacité de détecter les phénomènes diminue. On retrouve dans la littérature deux modèles de détection [15] : les modèles de détection déterministes et les modèles de détection probabilistes. Ces modèles sont basés sur la probabilité de détection [31].

1.8.5.1 Modèle de détection déterministe (binaire)

Le modèle le plus utilisé dans la littérature est le modèle de détection binaire. Il a l'avantage d'être rapide, simple à implémenter et il a un faible coût de calcul. Il est aussi appelé modèle 0-1. Si un évènement se produit dans le Rd du capteur alors il est

détecté à 100% (probabilité de détection = 1). Par contre, si celui-ci se produit à une distance supérieure au rayon Rd, la probabilité de le détecter est nulle (Pd =0).la figure (1.8) illustre ce modèle de détection d'événement. D'autres modèles plus réalistes intègrent le fait que la probabilité de détection d'un capteur diminue avec l'accroissement de la distance. Il s'agit des modèles de détection probabilistes que nous allons exposer dans la section suivante.

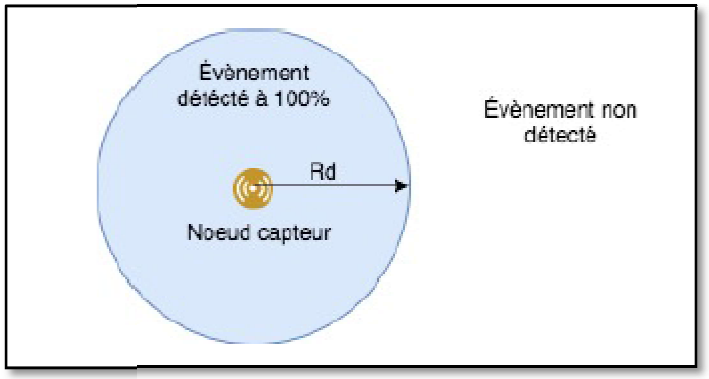


Figure 1.8 : Modèle de détection binaire

1.8.5.2 Modèle de détection probabiliste

Les modèles asymptotiques dépendent de la distance. En effet, lorsqu'un évènement se produit près du capteur, il a une plus grande probabilité d'être détecté. Cette probabilité diminue en s'éloignant du capteur. On distingue deux fonctions pour le calcul de la probabilité de détection : exponentielle, polynomiale.

Modèle exponentiel : Dans ce modèle, la probabilité de détection d'un évènement diminue exponentiellement par rapport à la distance. La probabilité de détection est donnée par la fonction suivante :

$$P_s^p = \exp(-\alpha d(s,p)) \qquad 1.6$$

Où α représente la qualité de détection du capteur *s* et *d(s,p)* la distance euclidienne entre le capteur s et la position *p* [16].

Modèle polynomial : Dans ce modèle, la probabilité de détection d'un évènement se dégrade de manière polynomiale d'ordre (-*k*). Elle est donnée par la fonction suivante :

$$P_s^p = \lambda d(s,p)^{-k} \qquad 1.7$$

où *k* représente la qualité de détection du capteur et λ est lié aux configurations matérielles de celui-ci. Une combinaison du modèle de détection binaire et du modèle de détection asymptotique est apparue. Ce nouveau modèle considère deux zones de détection : une zone de confiance et un cercle maximal. Comme on peut le voir dans la figure 1.9, on distingue 3 zones :

Zone (1) : appelée zone de confiance. Si l'évènement se produit dans cette zone, il est détecté avec une probabilité égale à 1.

Zone (2) : située entre la zone de confiance et la limite maximale de détection. Un évènement est détecté avec une probabilité Pp (qui peut être une fonction exponentielle ou polynomiale). Cette probabilité diminue avec la distance.

Zone (3) : située en dehors du champ de détection du capteur, au-delà de la limite maximale. Si un évènement se produit dans cette région, il ne sera pas détecte.

Ce modèle est représente par les équations suivantes :

$$P_p = \begin{cases} 1 & \text{si } d(s,p) \leqslant R_{min} \\ \exp(-\alpha d(s,p)) & \text{si } R_{min} \leqslant d(s,p) \leqslant R_{max} \\ 0 & \text{si } d(s,p) \geqslant R_{max} \end{cases} \quad 1.8$$

$$P_p = \begin{cases} 1 & \text{si } d(s,p) \leqslant R_{min} \\ \frac{\lambda}{d(s,p)^k} & \text{si } R_{min} \leqslant d(s,p) \leqslant R_{max} \\ 0 & \text{si } d(s,p) \geqslant R_{max} \end{cases} \quad 1.9$$

Ou *Rmin* désigne le rayon de la zone de confiance, *Rmax* la limite maximale [16].

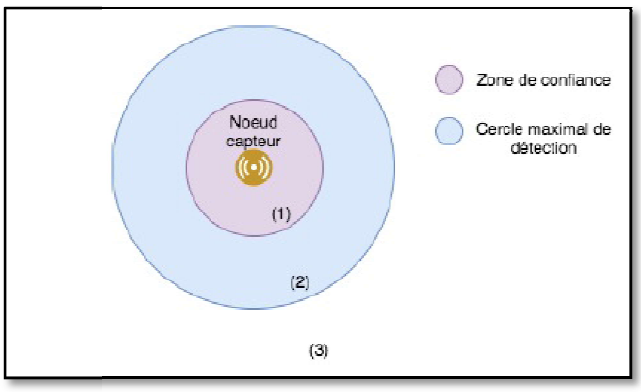


Figure1.9 : Modèle de détection probabiliste.

1.8.5.3 Modèle retenu

En utilisant le modèle de détection binaire et dans le but d'assurer une couverture complète de la région d'intérêt la matrice de couverture peut être définie comme suit :

$$Couv(i,j) = \begin{cases} 1 & \text{si la cellule (i,j) est couverte par au moins un capteur ;} \\ 0 & \text{sinon.} \end{cases} \quad 1.10$$

Le taux de couverture peut être calculé selon l'équation suivante :

$$Couv_{RI} = \frac{\sum_{i=0}^{l-1}\sum_{j=0}^{L-1} Couv(i,j)}{L \cdot l} \cdot 100 \quad 1.11$$

1.8.6 Connectivité

La connectivité est un paramètre essentiel qu'il faut prendre en considération lors du déploiement des RCSFs. Elle permet au capteur de transmettre les données vers la station de base. La transmission des données peut se faire directement entre le capteur et la station de base ou bien à travers d'autres nœuds relais (communication multi saut).

Un réseau est dit connecté si chaque nœud déployé est connecté à au moins un nœud voisin. Si une liaison entre deux nœuds se rompt, cela peut entraîner une défaillance dans la communication. Pour palier à ce genre d'incident, on a recours au concept de la k-connectivité. Chaque capteur est connecté à au moins k autres capteurs figure 1.10. Ainsi la communication est assurée même si k-1 liens sont défaillants [1].

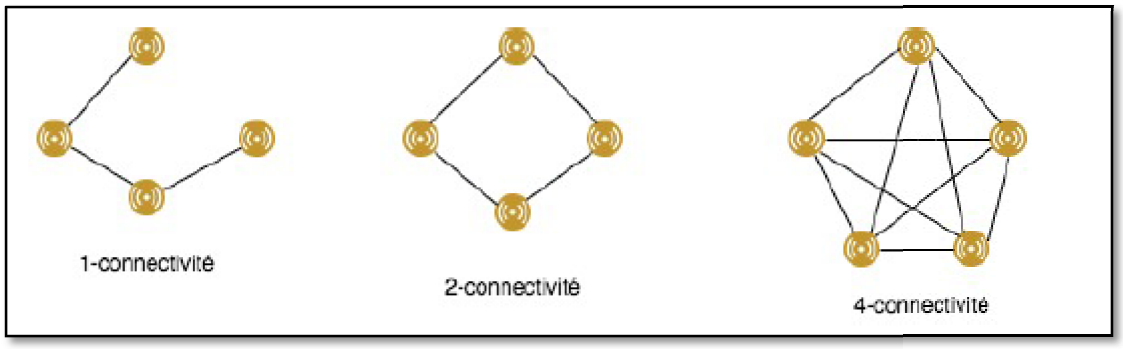


Figure1.10 : k-connectivité.

La connectivité d'un réseau est liée à la propagation d'ondes radios. Un nœud A est dit connecté à un autre nœud B si le RSSI du signal émis par le nœud B est supérieur à la sensibilité de l'antenne de réception du nœud A. 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.

1.8.6.1 Modèle de FRIIS

Le modèle de FRIIS est l'un des premiers modèles établies dans la littérature. Ce modèle, aussi appelle équation de transmission de *FRIIS*, exprime le rapport entre la puissance du signal reçu et celui émis par deux antennes séparées d'une distance R. Ces antennes sont considérées isotropes. Le rapport de puissance est donné par l'équation suivante :

$$\frac{P_r}{P_t} = n * \left[\left(\frac{\lambda}{4\pi R} \right)^2 * G_t * G_r \right] \qquad 1.12$$

Où *Pr* et *Pt* représentent la puissance reçue et la puissance de transmission. *Gt* et *Gr* représentent respectivement le gain de transmission et le gain de réception de l'antenne. Le terme $\left(\frac{\lambda}{4\pi R}\right)^2$ est appelé 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).

1.8.6.2 Modèle Multi Wall

Le modèle de propagation le plus approprie aux environnements intérieurs est le MWM (Modèle Multi Wall). Il prend en compte les atténuations dues à la pénétration de l'onde entre l'émetteur et le récepteur dans les obstacles (murs, portes, étages). Il ne considère pas les pertes par réfraction et par diffraction [42]. Le modèle est représenté par l'équation suivante :

$$PL_{MWF}[dB] = FSL + 10 * n * \log_{10}(d) + \sum_{i=1}^{I}\sum_{k=1}^{K_{wi}}(L_{wik}) + \sum_{j=1}^{J}\sum_{k=1}^{K_{fj}}(L_{fjk}) \quad 1.13$$

$$RSSI[dBm] = P_{TX}[dBm] - PL_{MWF}[dBm] \quad 1.14$$

Où :

-*FSL* : désigne Free-Space Losses ;

-L_{wik} : est l'atténuation du k^{éme} mur de la catégorie i ;

-L_{fjk} : est l'atténuation du k^{éme} étage de la catégorie j ;

-K_{wi} : est le nombre des murs de la catégorie i ;

-K_{fj} : est le nombre d'étages de la catégorie j.

Ce modèle est facile à implémenter et nécessite un temps de calcul réduit par Rapport à d'autres modèles de propagation.

1.8.6.3 Modèle retenu

Afin de modéliser la connectivité nous allons adopter le modèle le plus simple et le plus utilisé dans la littérature [13]. A savoir le modèle de communication binaire sous forme de disque voir la figure 1.11. Comme nous pouvons le voir dans la figure, le rayon de communication Rc est supérieur au rayon de couverture Rd.

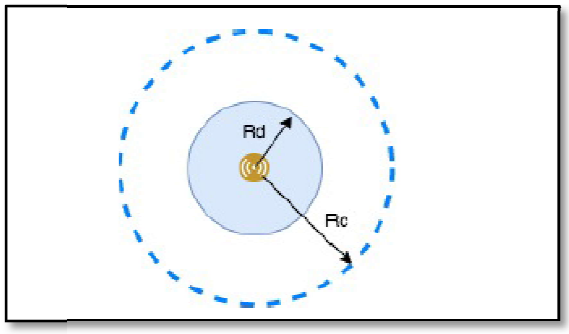


Figure1.11 : Rayon de communication d'un capteur

La matrice de connectivité est définie comme suit :

$$Con(i,j) = \begin{cases} 1 & \text{si la cellule (i,j) est dans le rayon} \\ & \text{de communication d'au moins 1 capteur;} \\ 0 & \text{sinon.} \end{cases} \quad 1.15$$

Le taux de connectivité peut être calculée _a partir de la matrice précédente à l'aide de l'équation :

$$Con_{RI} = \frac{\sum_i \sum_j Con(i,j)}{L * l} * 100 \quad 1.16$$

1.8.7 Durée de vie :

La durée de vie du réseau dépend principalement de la consommation d'énergie de ce dernier. En effet, les capteurs et les relais étant principalement alimentées par des batteries ont de ce fait une durée de vie limitée. Les composants radiofréquences sont plus gourmandes en énergie que l'unité de traitement des données. La durée de vie sera donc déterminée par la consommation en énergie de l'unité de communication qui dépend de la distance séparant les nœuds et du nombre de paquets à transmettre. Pour éviter les phénomènes de sur écoute, il sera préférable d'opter pour une communication directe entre les capteurs et les routeurs, dans une topologie en étoile [7].

1.9 Conclusion :

Dans ce chapitre, nous avons décrit et modélisé tous les critères qui ont une influence sur les performances d'un réseau de capteurs sans fils. Nous avons fait une revue de littérature des différentes modélisations existantes et nous avons sélectionné celles qui sont les plus adaptées à notre problématique.

Dans le chapitre suivant, nous allons exposer les différentes méthodes utilisées pour la résolution et l'optimisation du problème de déploiement 2D des RCSFs en environnement. Le but du chapitre 2 est de déterminer la méthode la plus adéquate qui puisse prédire l'emplacement des nœuds tout en respectant les contraintes de couverture et connectivite.

Chapitre II

2. Algorithmes génétiques

2.1 Introduction

Le problème de déploiement d'un réseau de capteurs sans fil consiste à déterminer une solution pour le placement d'un nombre minimum des capteurs de à l'intérieur d'une zone en faisant en sorte qu'ils puissent voir le maximum d'espace possible.

Afin de déterminer le meilleur moyen de résoudre et d'optimiser le déploiement d'un RCSF nous allons faire un tour d'horizon des méthodes de résolution qui existent dans la littérature.

Les algorithmes génétiques sont des algorithmes d'optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils ont été initialement développés par Joh Holland (1975), il décrit comment appliquer les principes de la théorie d'évolution naturelle sur les problèmes d'optimisation. Le principe de ces algorithmes s'inspire directement des lois de la sélection naturelle, décrites par Darwin. Il combine une stratégie de "survie des plus forts" avec un échange d'information aléatoire mais structuré. Pour un problème pour lequel une solution est inconnue.

Dans ce second chapitre, d'abord nous allons détailler la technique des algorithmes génétique, ensuite nous exposerons d'une manière générale les méthodes les plus utilisées pour chaque opérateur génétique à savoir la sélection, la mutation et le croisement et enfin nous expliquons l'algorithme génétique multi-objectif NSGA II.

2.2 Algorithmes évolutionnaires

Les algorithmes évolutionnaires (AEs) sont l'un des méthodes stochastiques, ils sont inspirés de la théorie de l'évolution proposée par Charles Darwin[19]. Cette théorie repose sur le principe de la sélection naturelle et stipule que les individus les plus adaptés à l'environnement ont plus de chance de survivre et de se reproduire. En effet, les algorithmes font évoluer une population d'individus qui représentent les solutions du problème traité. Les performances des individus sont évaluées à chaque itération à l'aide d'une fonction fitness. Les meilleurs individus seront sélectionnés pour subir des transformations (croisement, mutation) afin de générer des descendants.

d'une fonction fitness. Les meilleurs individus seront sélectionnés pour subir des transformations (croisement, mutation) afin de générer des descendants.

Les individus (ou une partie des individus) de la nouvelle population vont remplacer ceux de la population courante pour former une nouvelle générations d'individus. Ces opérations sont répétées jusqu' à atteindre un critère d'arrêt. Une architecture typique d'un algorithme évolutionnaires est représentée par la figure 2.1.

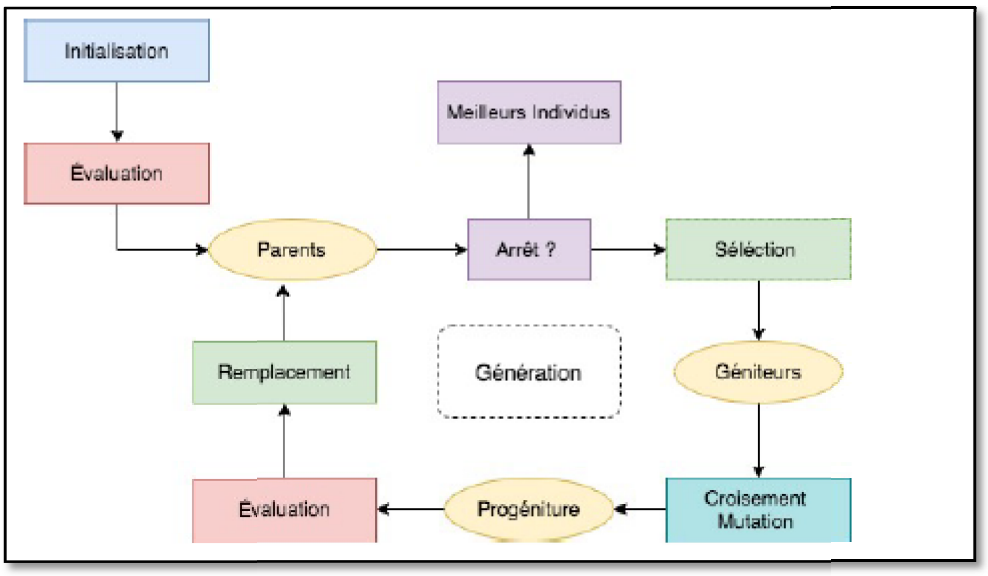


Figure 2.1:Organigramme d'un algorithme évolutionnaire

Il existe plusieurs variantes d'algorithmes évolutionnaires qui se sont développées indépendamment dans les années 60 et 70. Ces variant proposés au fil du temps se basent sur le même principe mais s'adaptent au problème traité. En effet, ces familles d'algorithmes différents uniquement par la structure du génotype des individus ou par les operateurs utilisés :

- Les stratégies d'évolution [20] : ces algorithmes permettent la résolution des problèmes d'optimisation continus. Ils sont dotés d'un operateur de mutation qui utilise une distribution normal. Il existe différentes variantes selon le nombre de descendant que l'on souhaite générer ou garder à chaque itération.
- La programmation évolutionnaire [21] : cette famille d'algorithmes est utilisée pour faire évoluer des automates à états finis. Chaque individu est considéré comme une espèce unique qui génère un descendant unique lors de l'étape de reproduction.

- Les algorithmes génétiques [22] : ce sont les algorithmes les plus connus et les plus utilisés. Ils sont caractérisés pas la représentation des individus sous forme de vecteur binaire ou chaîne de caractère. Nous nous intéresserons plus en détails.
- La programmation génétique [23] : cette famille d'algorithmes diffère des algorithmes génétiques uniquement dans la représentation des données. En effet, les individus sont représentes sous la forme d'un arbre plutôt qu'une chaîne.

2.3 Algorithmes génétique

John Holland, ses collègues et ses étudiants ont développé à l'université de Michigan les Algorithmes Génétiques (AGs) [24], métaphores biologiques inspirées des mécanismes de l'évolution darwinienne (sélection naturelle) et de la génétique. Ces métaphores prennent la forme d'algorithmes de recherche appelés "algorithmes génétiques" .Ces algorithmes font partie de la classe des algorithmes dits stochastiques. En effet une grande partie de leur fonctionnement est basée sur le hasard. Bien qu'utilisant le hasard, les AGs ne sont pas purement aléatoires. Ils exploitent efficacement l'information obtenue précédemment pour spéculer sur la position de nouveaux points à explorer, avec l'espoir d'améliorer la performance [25].

Les algorithmes génétiques permettent à une population de solutions de converger vers les solutions optimales. Pour ce faire, ils vont utiliser un mécanisme de sélection des individus de la population (les solutions potentielles). Les individus sélectionnés vont être croisés entre eux (exploitation), et certains vont être mutés (exploration). Ces mécanismes d'exploitation et d'exploration vont permettre de converger vers les bonnes solutions en évitant, autant que faire se peut, les optima locaux[25].

2.4 Terminologie et éléments de base

Un algorithme génétique recherche les extrêmes d'une fonction définie sur un espace de données appelé population. Par analogie avec la génétique, chaque individu de cette population est un chromosome et chaque caractéristique de l'individu est un gène. Dans un cas simple, un gène sera représenté par un bit (0 ou 1), un chromosome par une chaîne de bits. Chaque gène représente une partie élémentaire du problème, il peut être assimilé à une variable et peut prendre des valeurs différentes appelées allèles. La position du gène dans le chromosome se nomme locus [26].

On parle également de génotype et de phénotype. Le génotype représente l'ensemble des valeurs des gènes du chromosome alors que le phénotype représente la solution réelle

après transformation du chromosome. Lors de la génération d'une nouvelle population, des opérateurs génétiques tels que la sélection, le croisement et la mutation sont nécessaires pour la manipulation des chromosomes [26].

Le tableau (2.1) présente une récapitulation de la terminologie naturelle et celle utilisée par les algorithmes génétiques [24].

Nature	Algorithme génétique
Chromosome	Chaîne
Gène	Trait, caractéristique
Allèle	Valeur de la caractéristique
Locus	Position dans la chaîne
Génotype	Structure Ensemble des valeurs des gènes
Phénotype	Ensemble de paramètres, structure décodée Evaluation d'un génotype

Tableau (2.1) : Comparaison de la terminologie naturelle et celle des algorithmes génétiques

Les AGs utilisent donc un vocabulaire similaire à celui de la génétique. On parlera ainsi d'individus ou chromosomes dans une population. Chaque individu ou chromosome est constitué d'un ensemble d'éléments appelés gènes contenant les caractères héréditaires de l'individu. Ils utilisent un mécanisme de sélection naturelle, basée essentiellement sur la reproduction et sur le codage génétique qui stocke les informations décrivant l'individu sous forme de gènes [17] imitant les systèmes naturels de l'évolution des espèces.

2.5 Évolution des espèces

Dans un environnement quelconque dans lequel vit une population primitive, i.e. peu adaptée à cet environnement. Bien sûr, quoique globalement inadaptée, cette population n'est pas uniforme : certains individus sont mieux armés que d'autres pour profiter des ressources offertes par l'environnement (nourritures, abris, etc.) et pour faire face aux

dangers qui y rôdent (prédateurs, intempéries, etc.). Ces individus mieux équipés ont par conséquent une probabilité de survie plus grande que leurs congénères et auront de fait d'autant plus de chances de pouvoir se reproduire. En se reproduisant entre individus bien adaptés, ils vont transmettre à leurs enfants ces caractéristiques qui faisaient leur excellence. La population qui résultera de cette reproduction sera donc globalement mieux adaptée à l'environnement que la précédente puisque la plupart des individus auront hérité de plusieurs (puisque chacun hérite à la fois de sa mère et de son père) des caractéristiques de l' "élite" de la génération précédente. Et c'est ainsi, en recombinant à chaque génération les caractéristiques élémentaires de bonne adaptation et en saupoudrant-le tout d'un peu de hasard, que la population va évoluer vers une adéquation toujours meilleure avec l'environnement. Par analogie, les AGs joignent le même principe, ils sont basés sur le principe d'"évolution" d'une population d'individus. Dans celle-ci, ce sont en général les plus forts, c'est-à-dire les mieux adaptés au milieu, qui survivent et engendrent des progénitures. À partir des données du problème, on crée (généralement aléatoirement) une " population" de solutions admissibles. Puis on évalue chacune des solutions. On élimine une partie infime de celles qui se sont montrées inutiles, et on recombine les gènes des autres afin d'obtenir de nouveaux individus-solutions. Ainsi, à chaque génération un nouvel ensemble de créatures artificielles (des chaînes de caractères) est crée en utilisant des parties des meilleurs individus de la génération précédente ainsi que des parties innovatrices. Selon la théorie évolutionniste, cette nouvelle génération sera globalement plus adaptée au problème que la précédente. Ce procédé est alors répété jusqu'à la naissance d'une solution que l'on jugera satisfaisante [26].

2.6 Conception d'un algorithme génétique

La simplicité de mise en œuvre et l'efficacité constituent deux des caractéristiques les plus attrayantes de l'approche proposée par les AGs. La mise en œuvre d'un algorithme génétique sollicite la disponibilité [24][26] :

✓ d'une représentation génétique du problème, c'est-à-dire un codage approprié des solutions sous la forme de chromosomes. Cette étape associe à chacun des points de l'espace de recherche une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité. La qualité du codage des données conditionne le succès des algorithmes génétiques

✓ d'un mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population non homogène qui servira de base pour les générations futures.

Le choix de la population initiale est important car il peut prendre plus ou moins rapidement la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien sur le problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche ;

✓ d'une fonction d'évaluation pour mesurer la force de chaque chromosome ;

✓ d'un mode de sélection des chromosomes à reproduire ;

✓ des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace de recherche. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace de recherche ;

✓ des valeurs pour les paramètres qu'utilise l'algorithme : taille de la population, nombre total de générations ou critère d'arrêt, probabilités de croisement et de mutation.

2.7 Comment fonctionnent l'algorithme génétique ?

Un algorithme génétique fonctionne typiquement à travers un cycle simple de quatre étapes [26]:

1. création d'une population de chromosomes ;
2. évaluation de chaque chromosome ;
3. sélection des meilleurs chromosomes ;
4. manipulation génétique, pour créer une nouvelle population de chromosomes.

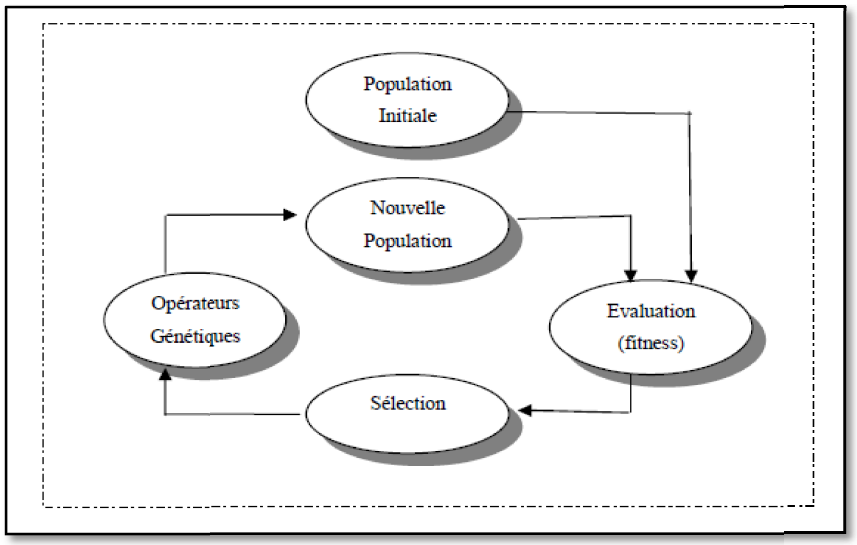


Figure 2.2:Cycle génétique

Le cycle décrit par la figure (2.2) est inspiré par la terminologie génétique. Lors de chaque cycle, une nouvelle génération de solutions du problème est obtenue.
Initialement, une population initiale est générée où chaque individu-solution de la population est codé sous forme d'une chaîne de caractères (chromosomes). Ensuite, une évaluation de chaque chromosome sera établie. Cette évaluation consiste à évaluer la qualité des chromosomes à l'aide de la fonction d'évaluation : fitness. Ce qui permet de sélectionner les chromosomes les plus adaptés et par conséquent leur appliquer les opérateurs génétiques (croisement et mutation) ce qui crée une nouvelle génération.

A la fin du cycle, une nouvelle population est acquise ouvrant ainsi la voie pour une nouvelle génération et par conséquent un nouveau cycle.

2.8 Variantes

En fait, les algorithmes génétiques sont une famille d'algorithmes, basés autour des mêmes idées. Cependant il existe beaucoup de variantes possibles suivant la représentation choisie, les opérateurs de croisement, de mutation et de sélection [17]. La section suivante présente les choix les plus courants qui définissent les variantes.

2.8.1 Codage

Le codage est une modélisation d'une solution d'un problème donné sous forme d'une séquence de caractères appelée chromosome où chaque caractère, dit aussi gène, représente une variable ou une partie du problème. La tâche principale consiste à choisir le contenu des gènes qui facilite la description du problème et respecte ses contraintes [26]. La littérature définit deux types de codage : binaire et réel.

2.8.1.1 Codage binaire

Le codage classique utilise l'alphabet binaire : 0,1. Dans ce cas le chromosome représente simplement une suite de 0 et de 1. Le codage binaire est également indépendant des opérateurs génétiques (croisement et mutation) du moment où ces derniers ne nécessitent aucune spécification. En effet, toute manipulation d'un chromosome donne naissance à un nouveau chromosome valide. Dans la pratique, le codage binaire peut présenter des difficultés. En effet, il est parfois très difficile ou très lourd de coder des solutions de cette manière. En outre, dans certain cas la taille mémoire requise peut devenir prohibitive [26].

2.8.1.2 Codage réel

Pour certain problème d'optimisation, il est plus pratique d'utiliser un codage réel des chromosomes. Un gène est ainsi représenté par un nombre réel au lieu d'avoir à coder les réels en binaire puis de les décoder pour les transformer en solutions effectives. Le codage réel permet d'augmenter l'efficacité de l'algorithme génétique et d'éviter des opérations de décodage supplémentaires. En effet, un chromosome codé en réels est plus court que celui codé en binaire.

2.8.2 Évaluation : fitness

L'opérateur d'évaluation n'est pas anodin. Il est utilisé par l'opérateur de sélection pour faire son choix des individus à conserver. Ainsi, pour mesurer les performances de chaque individu qui correspond à une solution donnée du problème à résoudre, on introduit une fonction d'évaluation. Elle permet de quantifier la capacité d'un individu à survivre en lui affectant un poids couramment appelé fitness. La force de chaque chromosome de la population est calculée afin que les plus forts soient retenus (étape de sélection) puis modifiés (croisement et mutation). La complexité de la fonction d'évaluation dépend essentiellement du problème et de ses contraintes [25] [26].

Ces deux derniers éléments, codage et évaluation, sont les seuls éléments spécifiques au problème à résoudre. Une fois qu'ils sont fixés, l'algorithme génétique que l'on appliquera sera toujours le même [30].

2.8.3 Population initiale

Une fois le codage choisi, une population initiale formée de solutions admissibles du problème doit être déterminée. Plusieurs mécanismes de génération de la population initiale sont utilisés dans la littérature [26]. Le choix de l'initialisation se fera en fonction des connaissances que l'utilisateur a sur le problème. S'il n'a pas d'informations particulières, alors une initialisation aléatoire, la plus uniforme possible afin de favoriser une exploration de l'espace de recherche maximum, sera la plus adaptée. Mais dans d'autres cas, il est possible d'utiliser d'autres mécanismes. Par ailleurs, cette étape présente un problème principal qui est celui du choix de la taille de la population. En effet une population trop grande augmente le temps de calcul et demande un espace mémoire considérable, alors qu'une population trop petite conduit à l'obtention d'un optimum local.

2.8.4 Critère d'arrêt

Déterminer l'arrêt d'un processus génétique est l'une des difficultés majeures de l'approche génétique. En effet, si l'on excepte le cas des problèmes artificiels, on ne sait jamais si l'on a trouvé l'optimum. Dans la pratique, l'utilisateur déclare un nombre de générations maximum. La recherche peut également être stoppée lorsque tous les individus d'une même population sont des copies d'un même individu. On dit alors qu'il y a " perte de diversité génétique " [29]. Les critères d'arrêt se résument alors en :

1. Arrêt après un nombre de générations fixé à priori.
2. Arrêt lorsque la population cesse d'évoluer ou en présence d'une population homogène.

2.8.5 Sélection

L'opérateur de sélection est chargé de " favoriser" les meilleurs individus [17]. Plus formellement, l'opérateur de sélection va générer à partir de la population courante une nouvelle population par copie des individus choisis de la population courante. La copie des chaînes s'effectue en fonction des valeurs de la fonction d'adaptation. Ce procédé permet de donner aux meilleures chaînes, une probabilité élevée de contribuer à la génération suivante. Cet opérateur est bien entendu une version artificielle de la sélection naturelle, la survie darwinienne des chaînes les plus adaptées [24].

Il existe de nombreuses techniques de sélection, les plus courantes seront évoquées dans la section suivante :

- **La sélection par classement** : elle consiste à ranger les individus de la Population dans un ordre croissant (ou décroissant selon l'objectif) et à retenir un nombre fixé de génotypes. Ainsi, seuls les individus les plus forts sont conservés. L'inconvénient majeur de cette méthode est la convergence prématurée de l'algorithme génétique. Il est parfois nécessaire de garder quelques individus jugés faibles pour créer la diversité au niveau de la population. Une autre difficulté consiste à fixer une limite à la sélection ce qui empêche parfois de garder des bons candidats pour les futures générations [26].
- **La sélection par la roulette** : elle consiste à créer une roue de loterie biaisée pour laquelle chaque individu de la population occupe une section de la roue proportionnelle à sa valeur d'évaluation. Ainsi, même les individus les plus faibles ont une chance de survivre.

Si la population d'individus est de taille égale à *N*, alors la probabilité de

sélection d'un individu xi notée *p(xi)* est égale à :

$$P(x_i)=\frac{F(x_i)}{\sum_{k=1}^{N} F(x_k)} \qquad (2.1)$$

En pratique, on calcule pour chaque xi individu sa probabilité cumulée

$$q_i = \sum_{j=1}^{i} p(x_j) \qquad (2.2)$$

et on choisi aléatoirement un nombre *r* compris entre 0 et 1.

L'individu retenu est x_1 si $q_1 \geq r$ ou xi $(2 \leq i \leq N)$ si $q_{i-1} < r \leq q_i$. Ce processus est répété N fois. Avec une telle sélection, un individu fort peut être choisi plusieurs fois. Par contre, un individu faible à moins de chance d'être sélectionné [26].

C'est cette sélection qui a été exclusivement utilisée dans ce mémoire.

- **La sélection par tournoi** : elle consiste à choisir aléatoirement deux ou plusieurs individus et à sélectionner le plus fort. Ce processus est répété plusieurs fois jusqu'à l'obtention de N individus. L'avantage d'une telle sélection est d'éviter qu'un individu très fort soit sélectionné plusieurs fois [26].

2.8.6 Croisement

La naissance d'un nouvel individu, nécessite la prise aléatoire d'une partie des gènes de chacun des deux parents. Ce phénomène, issu de la nature est appelé croisement (crossover). Il s'agit d'un processus essentiel pour explorer l'espace des solutions possibles. Une fois la sélection terminée, les individus sont aléatoirement répartis en couples. Les chromosomes parents sont alors copiés et recombinés afin de produire chacun deux descendants ayant des caractéristiques issues des deux parents.

Dans le but de garder quelques individus parents dans la prochaine population, on associe à l'algorithme génétique une probabilité de croisement, qui permet de décider si les parents seront croisés entre eux ou s'ils seront tout simplement recopiés dans la population suivante [26].

La littérature définit plusieurs opérateurs de croisement. Ils diffèrent selon le type de codage adapté et la nature du problème traité.

2.8.6.1 Croisement binaire

Ce croisement peut avoir recourt à plusieurs types en occurrence [26]

- **Croisement en 1-point :** c'est le croisement le plus simple et le plus connu dans la littérature. Il consiste à choisir au hasard un point de croisement pour chaque couple de chromosomes. Les sous-chaînes situées après ce point sont par la suite inter-changées pour former les deux fils figure (2.3).

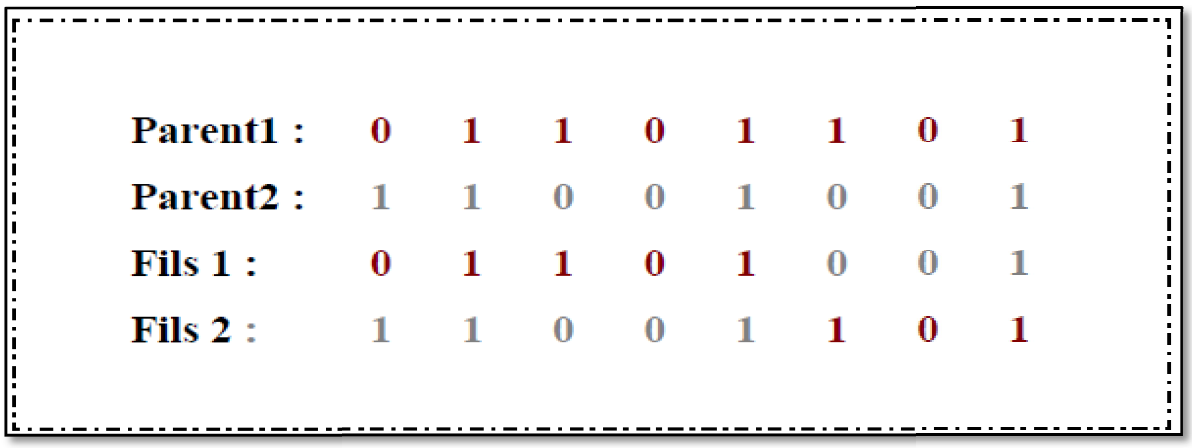


Figure 2.3:Croisement en un point de deux chromosomes

- **Croisement en n-points** : Ce type de croisement s'énonce par un choix aléatoirement de n-points de coupure pour dissocier chaque parent en *n+1* fragments. Pour former un fils, il suffit de concaténer alternativement *n+1* sous chaînes à partir des deux parents. Ce croisement cherche à explorer tout l'espace de solutions possibles en créant des descendants ayant des caractéristiques très loin des parents.

- **Croisement en 2-points :** c'est un cas particulier du croisement en n-points. On choisit aléatoirement deux points de coupure pour créer les descendants.

- **Croisement uniforme** : cette technique génère des progénitures gène par gène à partir des deux parents. Il existe des versions distinctes de ce croisement. La plus connue est celle qui utilise un masque. S'il est égal à 1, l'enfant 1 reçoit l'allèle correspondant du parent 1 et l'enfant 2 reçoit celui du parent 2. Sinon, l'échange se fait dans l'autre sens Figure (2.4).

Parent1 :	0	1	1	0	1	1	0	1
Parent2 :	1	0	0	0	1	0	1	1
Masque	0	1	0	1	0	0	1	1
Fils1 :	1	1	0	0	1	0	0	1
Fils2 :	0	0	1	0	1	1	1	1

Figure 2.4:Croisement uniforme

2.8.6.2 Croisement réel

Le codage réel requiert des opérateurs génétiques spécifiques pour la manipulation des chromosomes. Il est de plusieurs types [26] :

- **Ordre de base cyclique** : pour créer un fils, il suffit de copier une sous-chaîne d'un parent et de compléter les gènes manquants à partir de l'autre parent, en maintenant l'ordre des gènes. Généralement, une fois deux chromosomes parents sélectionnés pour le croisement, deux points de coupures sont choisis aléatoirement sur chaque parent. Ensuite on place les sous-chaînes entre les points de coupure sur les deux fils dans la même position que les parents. Pour compléter les gènes manquants du fils 1, on commence par insérer les gènes situés à droite du deuxième point de coupure du parent 2 tout en gardant l'ordre des gènes et en ignorant les gènes déjà pris. Le deuxième fils est complété à partir du parent 1 de la même manière que le fils 1. La Figure (2.5) montre sur un exemple des étapes de ce type de croisement.

			1er pt				2ème pt	
Père 1	a	b	c	d	e	f	g	h	i
Père 2	f	b	g	a	e	i	c	h	d
Fils 1	.	.	.	d	e	f	g	.	.
Fils 2	.	.	.	a	e	i	c	.	.
Fils 1	a	i	c	d	e	f	g	h	b
Fils 2	d	f	g	a	e	i	c	h	b

Figure 2.5: Croisement d'ordre de base cyclique

- **Croisement uniformément continu** : ce type a été suggéré pour produire des chromosomes valides. Un chromosome X=(x1,x2,...,xn) est valide lorsque :

$$\sum_{i=1}^{n} x_i = 1 . \qquad (2.3)$$

Étant donné deux chromosomes valides X=(x1,x2,…..,xn) et les descendants et Y=(y1,y2,…….,yn) X'=(x'1,x'2……..,x'n) Y'=(y'1, y'2 ,…….y'n) sont définis de la façon suivante : x'i =sxi +(1−s)yi et y'i=(1−s)xi+syi . Où s est une constante choisie à chaque itération aléatoirement dans l'intervalle[−0.5,0.5].

- **Croisement d'ordre maximal** : ce type de croisement a pour objectif de garder le maximum possible les positions et l'ordre des gènes. On commence par choisir aléatoirement deux points de coupure. Les sous-chaînes situées au milieu sont inter-changées. Les gènes manquants sont par la suite complétés à partir de chaque père en allant de gauche à droite et en choisissant le premier caractère disponible. A la différence du croisement de base cyclique, le fils 1 est complété à partir du parent 1 et le fils 2 à partir du parent 6. La Figure 5 illustre un exemple de croisement d'ordre maximal.

Père 1 :	a	b	c	d	e	f	g	h	i
Père 2 :	f	b	g	a	e	i	c	h	d
Fils 1 :	b	d	f	a	e	i	c	g	h
Fils 2 :	b	a	i	d	e	f	g	c	h

Figure 2.6:Croisement d'ordre maximal

2.8.7 Mutation

La mutation est définie étant la modification aléatoire d'une partie d'un chromosome. Elle constitue une exploration aléatoire de l'espace des chaînes [24].C'est un phénomène qui a un rôle théoriquement plus marginal : il est là pour éviter une perte irréparable de la diversité. Différentes manières de mutation d'un chromosome sont aussi définies dans la littérature.

2.8.7.1 Mutation en codage binaire

Dans un algorithme génétique simple, la mutation en codage binaire est la modification aléatoire occasionnelle (de faible probabilité) de la valeur d'un caractère de la chaîne [30].

2.8.7.2 Mutation en codage réel

Pour le codage réel, les opérateurs de mutation les plus connus et les plus utilisés sont les suivants [26] :

- L'opérateur d'inversion simple : consiste à choisir aléatoirement deux points de coupure et inverser les positions des bits situés au milieu.
- L'opérateur d'insertion : consiste à sélectionner au hasard un bit et une position dans le chromosome à muter, puis à insérer le bit en question dans la position choisie.
- L'opérateur d'échange réciproque : cet opérateur permet la sélection de deux bits et les inter changés.
- L'utilisation de probabilités ne signifie pas que la méthode n'est qu'une exploration aléatoire. Les AGs utilisent des choix aléatoires comme des outils pour guider l'exploration à travers les régions de l'espace de recherche, avec une amélioration probable [24].

2.8.8 Valeurs des paramètres

Les paramètres qui conditionnent la convergence d'un algorithme génétique sont :

- ✓ la taille de la population d'individus ;
- ✓ le nombre maximal de générations ;
- ✓ la probabilité de croisement ;
- ✓ la probabilité de mutation.

Les valeurs de tels paramètres dépendent fortement de la problématique étudiée. Ainsi il n'existe pas de paramètres qui soient adaptés à la résolution de tous les problèmes qui peuvent être posés à un algorithme génétique. Cependant, certaines valeurs sont souvent utilisées (définies dans la littérature) et peuvent être de bons points de départ pour démarrer une recherche de solutions à l'aide d'un AG.

- ✓ la probabilité de croisement est choisie dans l'intervalle [0.1, 0.9] ;
- ✓ la probabilité de mutation est choisie dans l'intervalle [0.01, 0.1].

Trouver de bonnes valeurs à ces paramètres est donc un problème parfois délicat.

2.9 Applications les Algorithmes génétiques

Ayant été reconnue comme une approche valide des problèmes nécessitant une exploration performante et économique du point de vue calcul, les algorithmes génétiques sont maintenant appliqués plus largement, aux domaines des affaires, à la recherche scientifQ<ique en général, ainsi que pour l'industrie. Les raisons de ce nombre grandissant d'applications sont claires. Ces algorithmes sont simples d'un point de vue de calcul, cependant très performants dans leur recherche d'amélioration [24].

Les AGs peuvent traiter deux types de problèmes : les problèmes mono objectifs qui visent à optimiser un seul critère et les problèmes multi-objectifs qui visent à optimiser simultanément plusieurs critères.

2.10 Les algorithmes génétiques multi-objectifs

En 1993 Fonseca et Fleming ont proposé une méthode dans laquelle chaque individu de la population est rangé en fonction du nombre d'individus qui le dominent. Ensuite, ils ont utilisé une fonction de notation permettant de prendre en compte le rang de l'individu et le nombre d'individus ayant le même rang. Soit un individu xi à la génération t, dominé par pi(t) individus. Le rang de cet individu est :

$$\text{Rang}(x_i, t) = 1 + p_i(t) \tag{2.3}$$

Tous les individus non dominés sont de rang 1.Dans l'exemple ci-dessous les points 1,3 et 5 ne sont dominés par aucun autre .Alors que le point 2 est dominé par le point 1, et que le point 2 est dominés par le point 1, et que le point 4 est dominés par les point 3 et 5.

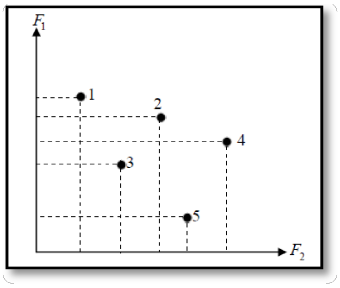


Figure 2. 7 : Exemple de dominance

Les étapes de l'algorithme génétique multi-objectif sont :

1. initialisation de la population.
2. L'évaluation

2.a. l'évaluation de la qualité de la solution en termes de la convergence

2.b. l'évaluation de la qualité de la solution en termes de la diversité

3. sélection.

4. croisement.

5. mutation.

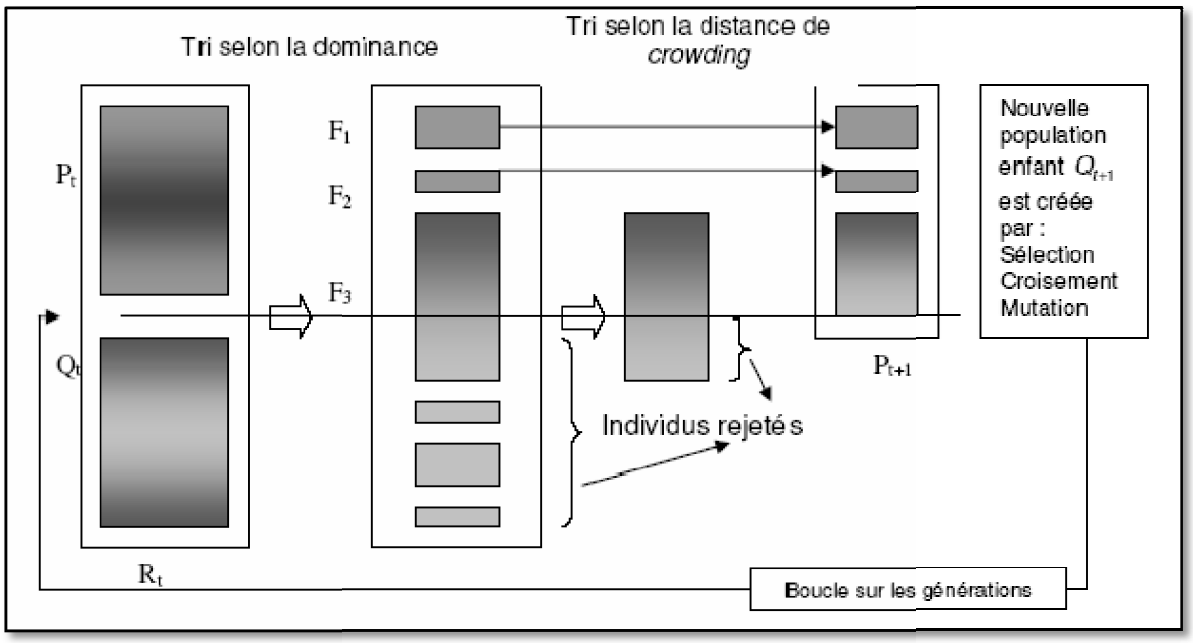
6. remplacement.

2.10.1 Rang de Pareto optimal

En absence d'information concernant l'importance des objectifs, la dominance de Pareto est la seule méthode de déterminer l'évaluation de performance. Donc tous les individus Non dominés sont considéré les meilleurs individus et ainsi sont assignés la même fitness, par exemple zéro. Cependant, la détermination d'une valeur de fitness pour les individus dominés est une question plus subjective. Les individus sont rangés, en un vecteur but et on fait la préférence (but, priorité). La considération du but et de la priorité sélective exclut des objectifs selon leurs priorités et s'ils atteignent aux leurs buts, l'information de priorité et de but peuvent être souvent extraite directement à partir de la description de problème. Les priorités sont des valeurs de nombre entier qui déterminent auxquels ordre d'objectifs doivent être optimisés, selon leur importance. Fréquemment dans le contrôle des systèmes, la stabilité en boucle fermé a la priorité la plus élevée et il est d'abord minimiser les valeurs de buts indiquent le niveau désiré de performance dans chaque dimension d'objectif. Le vecteur de but trace la région de l'échange où MOGA concentre son effort de calcule. Une fois le rang est trié, cet opérateur génétique assignera la fitness aux individus par l'interpolation du meilleur au plus mauvais, selon une règle exponentielle. Alors une valeur simple de fitness est dérivée pour chaque groupe d'individus avec le même coût, en utilisant la moyenne.

2.10.2 NSGA-II

NSGA II (Non-dominated sorting genetic algorithm) est une méthode proposée par Deb et al. [6]. Il s'agit d'un AGMO base sur le principe de dominance selon Pareto. L'algorithme commence par générer la population initiale composée de N individus. Ensuite, les individus sont classes selon la notion de dominance de Pareto. Les individus non-dominés sont regroupes dans le premier front F1. Le reste des individus sont Chapitre 4. Résolution du problème de déploiement d'un RCSF dans un environnement, ensuite classés et d'autres fronts apparaissent. Le principe de la méthode est présent dans la figure suivante :



Figure(2.8): Principe de fonctionnement de l'algorithme NSGAII.

Dans NSGA-II la population des enfants Qt est d'abord créée en utilisant la population des parents Pt. Les deux populations sont ensuite réunies pour former la population mixte Rt 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 F1 correspond à l'ensemble des solutions non-dominées de Rt. En considérant le reste des individus dans Rt après avoir enlever ceux de F1, et après avoir réaliser un nouveau tri de dominance, nous obtenons le deuxième front F2 constitué des individus non-dominés de l'ensemble (Rt\F1). Cette procédure est répétée jusqu'à ce que tous les individus de Rt soient attribués à un front. Par la suite, la nouvelle population Pt+1 est créée et est remplie au fur et à mesure avec les différents fronts successifs. Comme la taille de la population Rt 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 considéré dans le but d'améliorer la diversité des solutions. La procédure de NSGA-II est résumée dans l'algorithme suivant :

Algorithm 1 NSGA-II

$t \leftarrow 0$, $P_0 \leftarrow$ random(), $|P_0|=N$
repeat
  $Q_t \leftarrow$ variation (Select(P_t)). // sélection parentale :Tournoi de taille 2 ; variation : croisement, mutation.
  $R_t \leftarrow P_t \cup Q_t$
  $\mathcal{F} \leftarrow$ Tri-dominance(R_t) // $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, ...)$
  $P_{t+1} \leftarrow \emptyset$, $i \leftarrow 1$
  while $|P_{t+1}| + |\mathcal{F}_i| < N$ **do**
    $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i$; $i \leftarrow i + 1$
  end while
  $P_{t+1} \leftarrow$ Tri-surpeuplement(P_{t+1})
  $P_{t+1} \leftarrow P_{t+1}[0 : N]$ // choisir les N premiers éléments
  $t \leftarrow t + 1$
until Critère d'arrêt rencontré

2.11 Les travaux connexes d'optimisation le problème de déploiement des RCSFs

Le déploiement des réseaux de capteurs sans fils est un problème largement traité dans la littérature. Nous allons exposer quelques travaux effectuées dans le cadre de l'optimisation du dé ploiement des RCSFs.

✓ Hanaa ZainEldine et al. [31] ont traité le problème de déploiement aléatoire de capteurs en introduisant une technique de déploiement dynamique améliorée basée sur les AGs. Le but était de maximiser la couverture du réseau et de réduire son cout en minimisant le nombre de capteurs. Les résultats des simulations ont montré l'efficacité de la solution technique proposée.

✓ Yinggao Yue et al. [32] se sont intéressés à la connectivité et la couverture d'un réseau de capteur sans fils. L'objectif étant d'améliorer ces deux critères en utilisant l'algorithme de colonies d'abeilles artificielles. Les simulations ont prouvé que comparée à la distribution aléatoire des nœuds, la méthode proposée aboutit à un RCSF avec un taux de couverture et de connectivité plus élevé. La redondance des données est également réduite ainsi que le trafic du réseau ce qui améliore son efficacité.

✓ Ali afhantoloee et al. [33] ont proposé une méthode pour le déploiement 3D des capteurs dans un environnement indoor. Le but est de faciliter la mobilité de personnes handicapées. Ils ont utilisé un algorithme basé sur la diagramme de Voronoi pour améliorer la couverture du réseau de capteurs.

✓ Hanh et al. [34] dans leur article parue en 2017 ont traité le problème de déploiement d'un réseau ayant différents types de capteurs. Ils ont proposé une méthode basée sur l'optimisation par essaim particulaire. Les résultats des simulations ont montré que cette méthode donne des solutions de meilleure qualité comparé à d'autres méthodes. La méthode a également fait ses preuves en terme de temps de calcul et de vitesse de convergence.

✓ Mohamed Benatia et al. [35] traitent du problème de déploiement des nœuds capteurs ainsi que des nœuds relais. Ils ont étudié deux variantes d'algorithmes évolutionnaires : algorithme génétique classique et NSGA-II. Plusieurs simulations ont été effectuée afin de déterminer les meilleures valeurs des paramètres de base (taille de la population, probabilité de croisement, probabilité de mutation ...). Les chercheurs avaient pour but d'optimiser la couverture, la connectivité ainsi que le cout des réseaux de capteurs déployés dans un smart building.

✓ Tian et al. [36] se sont intéressés aux réseaux de capteurs sans fils ayant des nœuds à énergie limitée. Leur objectif était d'optimiser la couverture du réseau et de réduire le nombre de nœuds actifs. Ils ont utilisé des méthodes basées sur les algorithmes génétiques et l'algorithme de colonies de fourmis binaire. Ils sont arrivés à la conclusion que la combinaison des deux approches donne lieu à un algorithme ayant une précision plus élevée et une convergence plus rapide.

✓ Akbarzadeh et al. [3] ont proposé une adaptation de la méthode de la descente du gradient pour optimiser le positionnement et l'orientation des capteurs. Ils ont utilisé des modèles réalistes qui prennent en considération la topographie de l'environnement. Les résultats ont montré que leur méthode avait un cout de calcul inférieur comparé à deux autres méthodes et donnait des résultats de meilleure qualité.

✓ N. Aitsaadi et al. [38] ont utilisé une méthode pseudo-aléatoire basée sur la recherche Tabou afin de déterminer le nombre optimal de nœuds ainsi que leurs positions dans le but de couvrir en totalité la région surveillée. L'étude a également pris en considération la contrainte de durée de vie et la connectivité. Les résultats ont montré que la méthode surpasse plusieurs autres approches utilisées dans la littérature.

2.12 Conclusion

Ce chapitre a établi les fondations nécessaires à la compréhension de l'algorithme utilisé pour notre travail (algorithmes génétiques), avec leurs mécanismes et leur

puissance. Ces algorithmes classés parmi les méthodes stochastiques, s'inspirent de l'évolution génétique des espèces, plus précisément du principe de la sélection naturelle. C'est initialement la quête de robustesse qui a orienté vers ces méthodes, les systèmes naturels sont robustes, efficaces et performants. En reproduisant sous forme artificielle le principe naturel de l'algorithme de sélection de la meilleure adaptation, les chercheurs visent l'atteinte des mêmes performances. Après cette étude théorique, nous allons entamer dans le chapitre suivant, la conception de notre système qui va résoudre le problème de répartition des capteurs dans un réseau sans fil, en s'inspirant des performances fournit par les algorithmes génétiques NSGA II.

Chapitre III

3. Conception du système

3.1 Introduction

Le déploiement d'un RCSF est un problème très complexe, surtout quand le nombre de critères pris en considération est supérieur à 2. Nous allons dans cette partie développer une application qui rendra la tache facile aux ingénieurs qui sont confrontés à ce genre de problèmes dans leur travail.

Nous prendrons en considération les différents modèles retenus dans le premier chapitre concernant le coût, la connectivité, la couverture et la sur-couverture. Nous allons ensuite les intégrer dans la méthode d'optimisation multi-objective que nous avons choisie : NSGA-II.

La première étape du cycle de développement consiste à analyser et définir les besoins, cette analyse est nécessaire pour pouvoir commencer la phase de conception qui permet à réaliser un processus récursif, en passant d'un niveau d'abstraction vers un autre plus bas. Ce raffinement rend la construction d'un outil logiciel plus facile.

Nous procédons d'une démarche descendante comme suite :

- ✓ Présentation de schéma global,
- ✓ Identification des composants ou bien schéma détaillé,

3.2 Motivation et objectif du système

Le but de notre travail est d'optimiser le déploiement d'un réseau de capteur sans fil dans un environnement. Pour ce faire il est nécessaire de modéliser les différents paramètres qui ont une influence sur le coût ainsi que sur la qualité de service du réseau de capteur. Nous allons donc dans ce chapitre modéliser l'espace de déploiement, le coût de déploiement ainsi les différents objectifs à optimiser qui sont (La couverture, la sur-couverture, la connectivité), on a utilisé l'algorithme génétique basé sur NSGA II pour

réaliser un système qui permet de trouver un plan de répartition optimal pour les capteurs dans un immeuble.

3.3 Conception globale

La conception globale nous permet d'avoir l'architecture générale, c'est une sorte d'une boite noire ayant comme entrées les informations nécessaires à la manipulation ou des données du moteur de fonctionnement pour en fournir les résultats attendus. On peut schématiser la fonctionnalité générale de notre système par le schéma suivant :

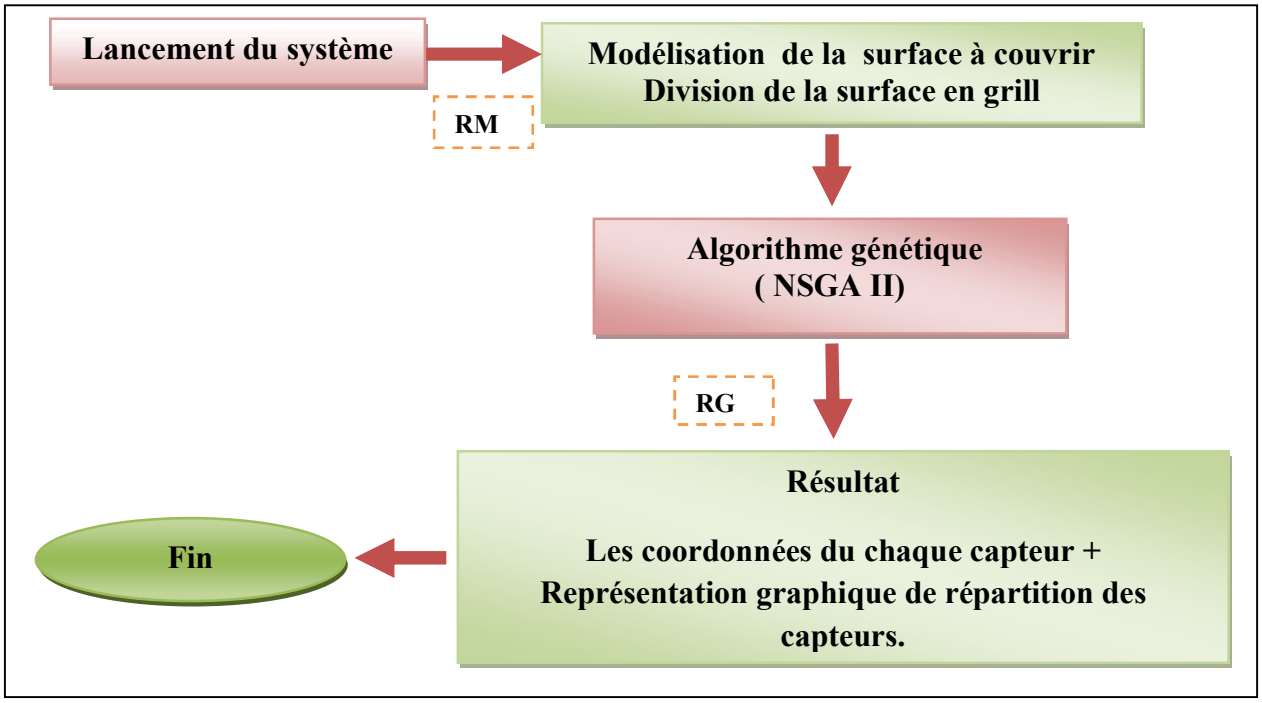


Figure 3.1:Conception globale du système

3.4 Conception détaillées

Cette phase est très indispensable afin de comprendre mieux le fonctionnement du système, car elle raffine le système en plusieurs composants pour permettre l'implémentation ultérieurs sans ambigüité, ainsi dans cette phase, nous montrons comment le travail doit être réalisé et dans quel ordre.

3.5 Les paramètres engendrés par les composants

Ce sont des paramètres manipulés par les modules du système lors de son déroulement.

- ✓ **La représentation matricielle (RM) :** c'est celle qui nous permet de représenter notre scène en fonction de leurs tailles et leurs architectures sous une matrice en termes de coordonnées adéquates, cette représentation a pour but de faciliter le suivi de fonctionnement du système.

- ✓ **La représentation graphique(RG) :** c'est la représentation de la scène qui contient les informations formelles (physique) de leurs composantes.

3.6 Modélisation de la surface à couvrir

Dans ce module, L'espace de déploiement est subdivisé en une grille, il en résulte un ensemble A (matrice) d'emplacements possibles (i,j) appartient à A pour les nœuds du réseau. Si un capteur est installé dans une position (i, j) de A, la variable Di,j est affectée de la valeur 1. Ceci se déroule comme ainsi :

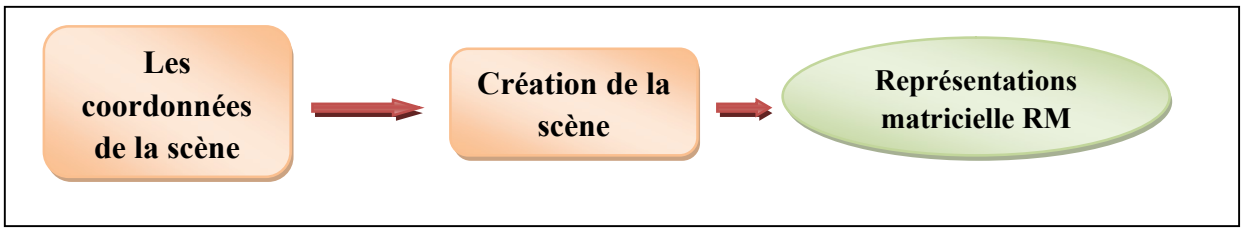


Figure 3.2:Modélisation de la surface à couvrir.

3.7 Algorithme génétique basé sur NSGA II:

Ce module est le noyau du travail à réaliser, elle représente l'implémentation de méthode de résolution des algorithmes évolutionnaires (algorithme génétique) multi-objectifs NSGA II. Le rôle est d'optimiser plusieurs critères (coût, couverture, connectivité, sur couverture) simultanément par l'exploitation de la structure des données implémentées dans le module précédent, ainsi que d'envoyer les résultats obtenus pour les afficher à l'utilisateur.

L'architecture suivante représente les étapes de l'algorithme.

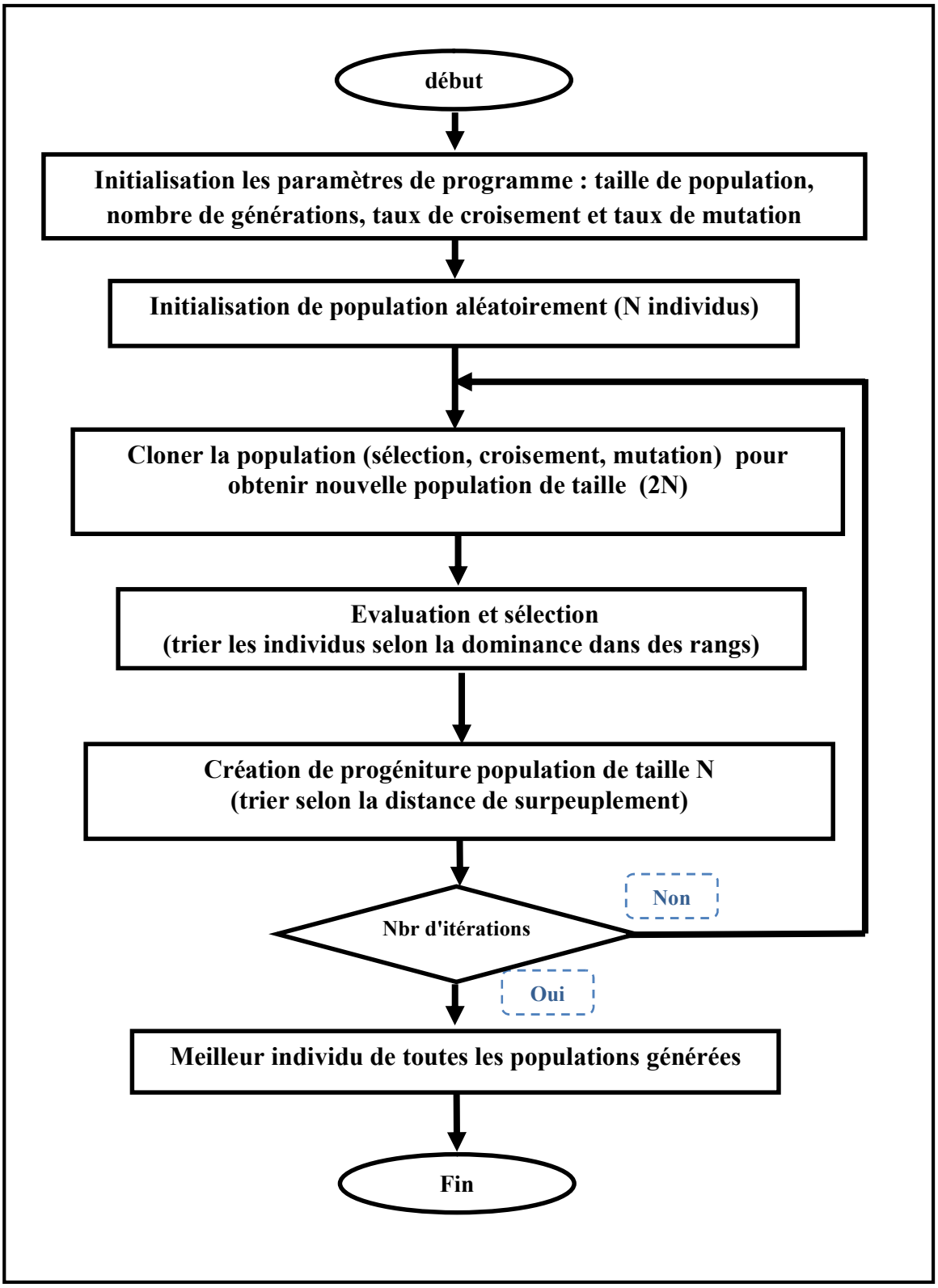


Figure 3.3:Architecture d'algorithme génétique (NSGA II).

3.7.1 Définition d'un individu (codage de chromosome):

L'espace de déploiement a été modélisé avec une grille qui est implémentée sous forme de matrice. Les algorithmes que nous avons sélectionnés ne peuvent manipuler ce type de structures de données. Il faut donc convertir cette matrice de déploiement en un vecteur comme on peut le voir dans la figure 3.4. Les individus qui sont les solutions du problème sont donc représentés sous forme de vecteur binaire. Chaque case porte deux informations : la présence ou l'absence d'un capteur, son emplacement.

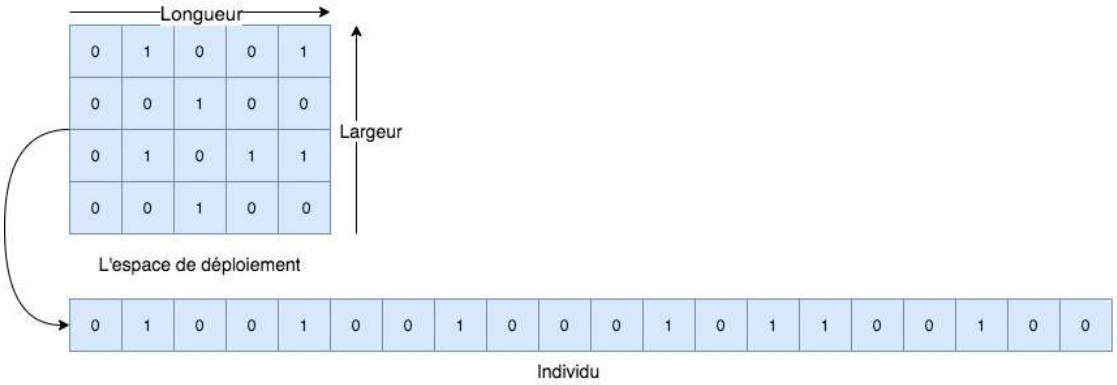


Figure 3.4:Codage d'un individu

3.7.2 Initialisation de population :

L'initialisation de la population se fait au début de l'exécution de l'algorithme. Après avoir défini la taille de la population, les algorithmes génèrent aléatoirement les individus. Le fait de générer les individus de manière aléatoire permet d'avoir une population de solutions diversifiées et évite que l'algorithme tombe dans un optimum local. La figure 3.5 montre un exemple d'une population de N individus.

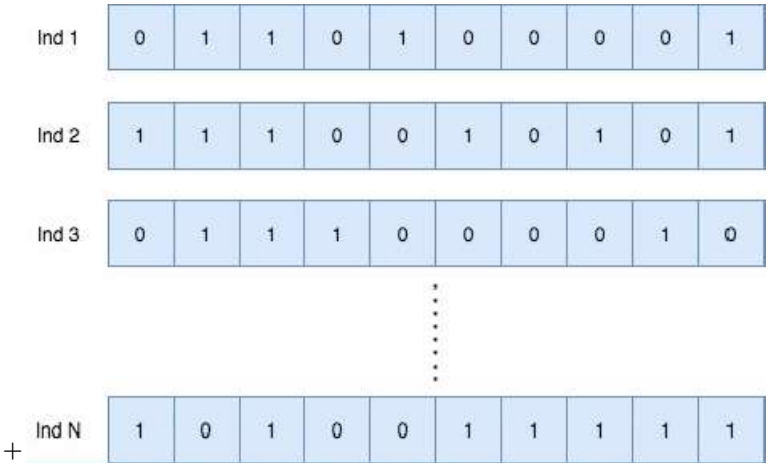


Figure 3.5:Population de N individus

3.7.3 Evaluation et sélection des individus

L'évaluation des individus X se fait à l'aide de la fonction fitness F (fonction d'évaluation). Dans notre cas la fonction d'évaluation évalue le coût de chaque individus, son taux de couverture, son taux de sur-couverture et enfin son taux de connectivité. La sélection des individus se fait selon la valeur fitness. Les individus ayant une meilleure valeur fitness sont sélectionnés et classés selon la notion de dominance de Paréto, Le critère de dominance selon Pareto est défini comme suit : Soit f_i , i ∈ [1, m] un ensemble de critères à minimiser, et x et x′ deux solution de l'espace réalisable. On dira que x domine x′ au sens de Pareto si ∀i∈ [1,m] f_i (x) ≤ f_i (x′), avec au moins une inégalité stricte. Une solution dite Pareto optimale est une solution non dominée, c'est à dire qu'aucune autre solution de l'espace réalisable ne domine cette solution, selon le critère de dominance.
Les objectifs à optimiser sont les suivant :

Min F1 = Coût (X)
Max F2 = Couv (X)
Max F3 = Con (X)
Min F4 = Sur-couv (X)

3.7.4 Choix de l'opérateur de croisement

Comme nous l'avons vu dans le chapitre précédent, il existe plusieurs types de croisement. Nous avons opté pour le croisement à 1-point dont le principe est représenté dans la figure (3.7) La probabilité de croisement désigne le nombre d'individus de la population (parents) qui vont subir l'opération de croisement pour former de nouveaux individus (enfants). Le point de croisement est choisi d'une manière aléatoire.

En prenant en compte les notes suivantes :

- ✓ **Héritabilité :** Les enfants doivent hériter les propriétés génétiques de leurs parents.
- ✓ **Validité** : les enfants (individus générés) doivent appartenir à l'espace de recherche.
- ✓ **La probabilité de croisement P_c** : probabilité de croiser deux parents, doit comprise entre 0.1 et 0.9.

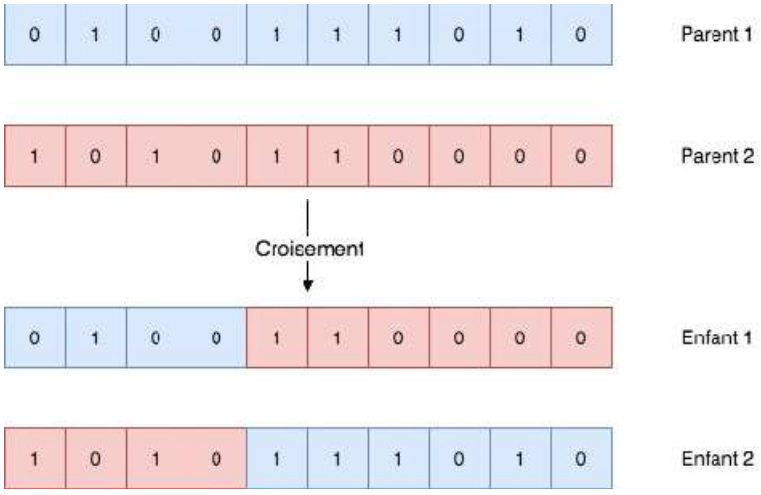


Figure 3.6:Croisement à 1 point

3.7.5 Choix de l'opérateur de mutation

L'opération de mutation est effectuée après l'opération de croisement. Cette Opération est réalisée avec un taux (probabilité de mutation) faible. Vu que nous avons opté pour une représentation binaire, nous ne pouvons utiliser un type de mutation autre que l'inversion binaire. Le principe de l'inversion binaire est représenté dans la figure 3.7. La mutation permet de générer des perturbations dans les solutions (individus) sélectionnés, il en résulte le maintien de la diversité de la population ainsi qu'une meilleure exploration de l'espace de recherche.

On prend en compte dans l'opération de mutation les notes suivantes :

- ✓ **Localité :** un changement local au niveau des coordonnées des capteurs.
- ✓ **Validité** : produire des solutions valides.
- ✓ **La probabilité de mutation (P_m)** : est généralement comprise entre 0.01 et 0.1

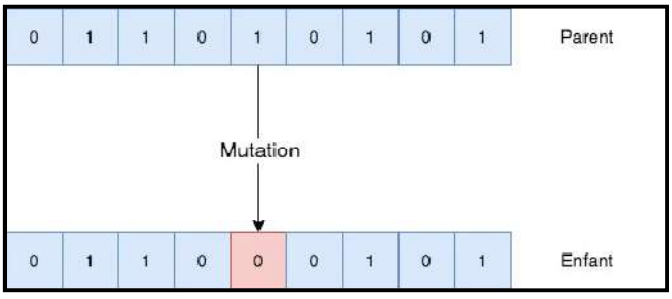


Figure 3.7 : Mutation à inversement de bit.

3.7.6 L'opérateur de remplacement

C'est la création d'une nouvelle population par l'évaluation (sélection, croisement et mutation) de population des parents et des enfants, et de classé selon la distance de surpeuplement (crowding) de façon qu'on choisir les N premier éléments et les restes seront éliminés.

3.7.7 Le critère d'arrêt

Il y'a deux critères d'arrêt dans Notre algorithme génétique, si l'une de deux critères est vérifié elle doit arrêter l'algorithme génétique.

- ✓ **Le premier critère :** si la valeur de fitness du meilleur individu de la population est stable pendant un nombre important d'itération (ne change plus), on peut dire que la solution trouvée est la solution optimale et on ne peut pas trouver une autre solution mieux que celle obtenu.
- ✓ **Le deuxième critère** : Si le nombre de génération atteint le nombre défini préalablement.

3.8 Conclusion

La conception nous a permis de voir le système en terme d'un ensemble de composants en interaction en partant d'un niveau d'abstraction élevé et en raffinant (la décompositions des modules en sous modules), pour aboutir a un niveau d'abstraction plus bas jusqu'à l'obtention d'une décomposition finale ou la génération du code est aisé. Le chapitre suivant va nous permettre d'implémenter les différents composants résultant de la phase de conception en choisissant un langage de programmation et en spécifiant les différents détails de chaque composant du système.

Chapitre IV

4. Implémentation et Résultats

4.1 Introduction

La réalisation d'un système doit être assurée par une étape finale dans son processus de développement, cette phase finale est considérée comme la phase de mise en œuvre du système attendu, pour que ce dernier réponde aux besoins soulignés dans le cadre de ce mémoire.

La conception de cette partie permet de mieux détailler les différentes notions théoriques envisagées tout au long des sections précédentes, elle permet aussi la réalisation des différentes parties du système tel qu'elles sont aboutis dans la conception. Pour cela, nous avons choisis un langage de programmation qui s'adapte avec notre type de système. D'autre part, nous allons présenter les algorithmes substantiels utilisés pour atteindre ce but.

Finalement, nous allons analyser les résultats obtenus à partir d'un ensemble des tests sur notre application

4.2 Le langage de programmation

4.2.1 Python

Python est le langage de référence pour plusieurs domaines y compris l'intelligence artificielle. Il existe sous une licence libre et fonctionne sur la plupart des plates-formes informatiques, des Smartphones aux ordinateurs, de Windows à Unix avec notamment GNU/Linux en passant par MacOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour augmenter la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe très clair et simple à utiliser ce qui induit à un temps de développement très court comparé a d'autres langages tel que JAVA, C++ ou Ruby. Il peut s'utiliser dans de nombreux contextes et s'adapter à plusieurs types d'utilisation grâce à des bibliothèques

spécialisées. Il est également doté d'une communauté très active. C'est donc en connaissance de cause que notre choix s'est porté sur Python.

4.2.2 PyCharm

Pour développer notre application, nous avons choisi un environnement de développement intégré (IDE : integrated development environment) python nommé Pycharm. Développé par l'entreprise tchéque JetBrains. Il est utilisé par des programmeurs professionnels dans le monde entier. Il s'agit d'un environnement multiplateforme qui fonctionne sur Windows, MacOS X et Linux. Il est disponible en trois versions, la version Community, la version éducationnel et la version Professional. Les deux premières versions sont open source, alors que la version professional est payante. La version community, celle que nous avons utilisé dans notre travail, possède différentes fonctionnalités tels que la coloration syntaxique, l'auto-complétion ou encore la vérification de code en direct. Cette version gratuite est très complète et répond à tous les besoins d'un développeur python [39].

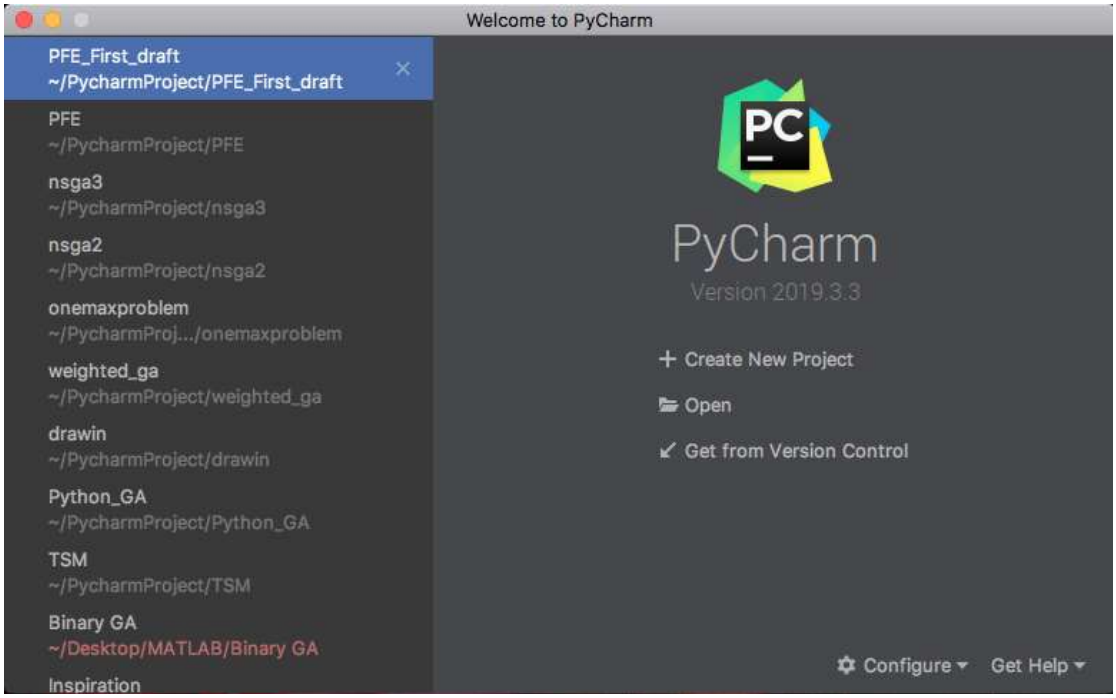


Figure 4.1:Composantes de l'IDE PyCharm

4.2.3 DEAP

DEAP (Distributed Evolutionary Algorithm in Python) [27] est un logiciel open source, sous licence LGPL, développé principalement au Laboratoire de Vision et

Systèmes Informatiques de l'Université Laval, Québec, Canada. Il a été conçu pour aider les chercheurs à développer des algorithmes évolutifs personnalisés. Il privilégie les algorithmes explicites et les structures de données transparentes, contrairement à la plupart des autres logiciels informatiques évolutifs qui ont tendance à encapsuler des algorithmes standardisés en utilisant l'approche de la boite noire. Sa boite à outils regroupe tous les opérateurs nécessaires et leurs arguments dans une structure pratique. Il comprend les fonctionnalités suivantes :

- ✓ Les algorithmes génétiques avec possibilité d'utiliser n'importe quelle structure de données (liste, tableau, ensemble, dictionnaire, arbre ...) ;
- ✓ La programmation génétique ;
- ✓ La stratégie d'évolution ;
- ✓ L'optimisation multi-objective (NSGA-II) ;
- ✓ La généalogie d'une évolution.

4.3 Construction des algorithmes

Nous allons dans cette partie expliquer les étapes d'algorithme que nous avons implémenté sous Python.

4.4 Importation des modules

Pour commencer, nous avons importé les modules et les bibliothèques contenant tous les outils nécessaires à l'exécution des algorithmes. Les modules "base, creator, tools" importé depuis la bibliothèque "deap" sont nécessaires à la définition des caractéristiques des algorithmes. Le module "random" permet de générer aléatoirement les individus d'une population. Enfin, le module "time" nous permettra d'avoir le temps d'exécution des simulations.

```
1  from deap import tools, base, creator
2  import random
3  import numpy as np
4  #import matplotlib
5  #matplotlib.use('TkAgg')
6  import matplotlib.pyplot as plt
7  import time
8  start = time.time()
9
```

4.5 Création des individus

La première étape est bien évidement la définition de l'espace de déploiement. Pour ce faire, l'application demande à l'utilisateur d'insérer la longueur ainsi que la largeur du l'environnement. Ensuite, la fonction fitness est créée. Comme nous avons 4 objectifs à optimiser nous avons introduit 4 poids dans la fonction : 1 pour maximiser une fonction, -1 pour la minimiser. à l'aide des outils "creator" et "toolbox" nous avons crée des individus sous forme de liste (vecteur) ayant comme valeur des 0 et des 1, d'une taille égale longueur * largeur du l'endroit. La population est bien crée maintenant.

```
9
10  # Les dimension de l'espace a déployé
11
12  length = int(input("entrer la longueure X"))
13  width = int(input("entrer la largeur Y "))
14  # creation de la fonction multi-objectifs
15  creator.create("fitnessmulti", base.Fitness, weights=(-1.0, 1.0, 1.0, -1.0))
16  # creation des individus
17  creator.create("individual", list, fitness=creator.fitnessmulti)
18  IND_SIZE = width * length
19  toolbox = base.Toolbox()
20  toolbox.register("attr_bool", random.randint, 0, 1)
21  toolbox.register("individual", tools.initRepeat, creator.individual, toolbox.attr_bool, n=IND_SIZE)
22  # la creation de la population
23  toolbox.register("population", tools.initRepeat, list, toolbox.individual)
24
```

4.6 Création de la fonction d'évaluation

Afin d'évaluer chaque individu, nous avons implémenté quatre fonctions objectifs à l'intérieur d'une seule et même fonction "evaluate" qui a pour argument la variable "individu". Pour ce faire, nous avons d'abord converti l'individu en forme de vecteur vers une forme matricielle. Puis nous avons extrait les coordonnées (i,j) des positions des capteurs. Ensuite nous avons définit le rayon de couverture des capteurs ainsi que leur rayon de connectivité .Ceci étant fait, nous procédons au calcul des matrices de couverture, connectivité et sur-couverture. Nous avons utilisé le principe de la distance euclidienne. Une cellule est dite couverte (reçois la valeur 1) si la distance la séparant du capteur est inférieur à son rayon de couverture. De même pour dire qu'une cellule se situe dans la zone de connectivité d'un capteur. Pour calculer la matrice de sur-couverture, nous parcourons la matrice de couverture et à chaque fois qu'une case porte une valeur supérieur ou égale à 2 (c'est à dire qu'elle est couverte

par 2 capteurs ou plus). La même case est affectée de la valeur 1 dans la matrice de sur-couverture. Les fonctions coût, couverture, connectivité et sur-couverture sont calcules suivant les formules qu'on peut voir dans les ligne 64,65,66 et 67 de l'algorithme. Elles sont ensuite retournées par la fonction "evaluate" pour permettre l'évaluation des individus par la fonction fitness.

```
64      a = sum(individual) # cout
65      b = ((sum(vect_a)) / IND_SIZE) * 100 # taux de detection de couverture
66      c = ((sum(vect_b)) / IND_SIZE) * 100 # taux de connectivite
67      u = ((sum(vect_c)) / IND_SIZE) * 100 # taux de sur-couverture
```

4.7 Définition des opérateurs génétiques

A l'aide de l'outil "toolbox.register", nous avons choisi les opérateurs à utiliser lors du processus d'évolution. Pour l'opération de croisement, nous avons choisi l'opérateur "cxOnePoint" (croisement à 1-point). Pour l'opération de mutation, nous avons choisi l'opérateur "mutFlipBit" (inversement de bit). Pour l'opération d'évaluation, nous avons choisi la fonction "evaluate" que nous avons créee au préalable. Enfin, en ce qui concerne l'opération de sélection, qui nous avons utilisée "selNSGA2" pour NSGA-II.

```
75  toolbox.register("mate", tools.cxOnePoint)
76  toolbox.register("mutate", tools.mutFlipBit, indpb=0.1)
77  toolbox.register("select", tools.selNSGA2)
78  toolbox.register("evaluate", evaluate)
79
```

4.8 Initialisation de la population :

Afin d'initialiser la population, il suffit de définir le nombre d'individus qu'elle contient (NPOP) et d'utiliser l'outil "toolbox.population". Ceci génèrent NPOP vecteurs auxquels il affecte aléatoirement des 0 et des 1 pour former les individus.

```
84
85      NPOP = 80
86      pop = toolbox.population(n=NPOP)
87
```

4.9 Processus d'évolution :

Avant d'entamer le processus d'évolution, il faut d'abord fixer les probabilités de mutation et de croisement. On commence par l'évaluation de la population initiale, puis on entre dans la boucle d'évolution. La première étape consiste en la sélection des individus qui formeront la prochaine génération (les descendants). Ces descendants sont ensuite soumis aux opérations de croisement et de mutation puis à l'opération d'évaluation, le but de ces opérations étant l'exploration de l'espace de recherche dans le but de trouver de meilleurs individus. Enfin, la population N-1 est remplacée entièrement ou en partie par les descendants. Quand la boucle atteint le nombre prédéfini de générations (itérations), le processus d'évolution s'achève et on obtient la population finale. L'outil de sélection permet d'ordonner la population finale selon le principe de sélection de chaque algorithme. On obtient ainsi une liste d'individus avec leur coût, taux de couverture, taux de connectivité et taux de sur-couverture. L'utilisateur peut choisir la solution qui lui convient le plus selon l'ordre de priorité qu'il donne aux différents critères de déploiement.

4.10 Simulation

4.10.1 Cas d'étude

Afin de tester et d'évaluer les performances des différents algorithmes, nous avons pris un cas d'étude. Il s'agit d'un espace restreint de $80m^2$ (un bureau) : 10 m de longueur et 8 m de largeur. voir les figures (4.2).Pour les paramètres de l'algorithme génétique nous avons fixé le nombre de générations à 50 et le nombre d'individu à 80,la probabilité de croisement à 0.9 et la probabilité de mutation à 0.1 et nous avons suppose que les capteurs avaient un rayon de couverture égale à 1.5m et un rayon de connectivité égale à 3m. en ensuite nous allons résoudre le problème de déploiement.

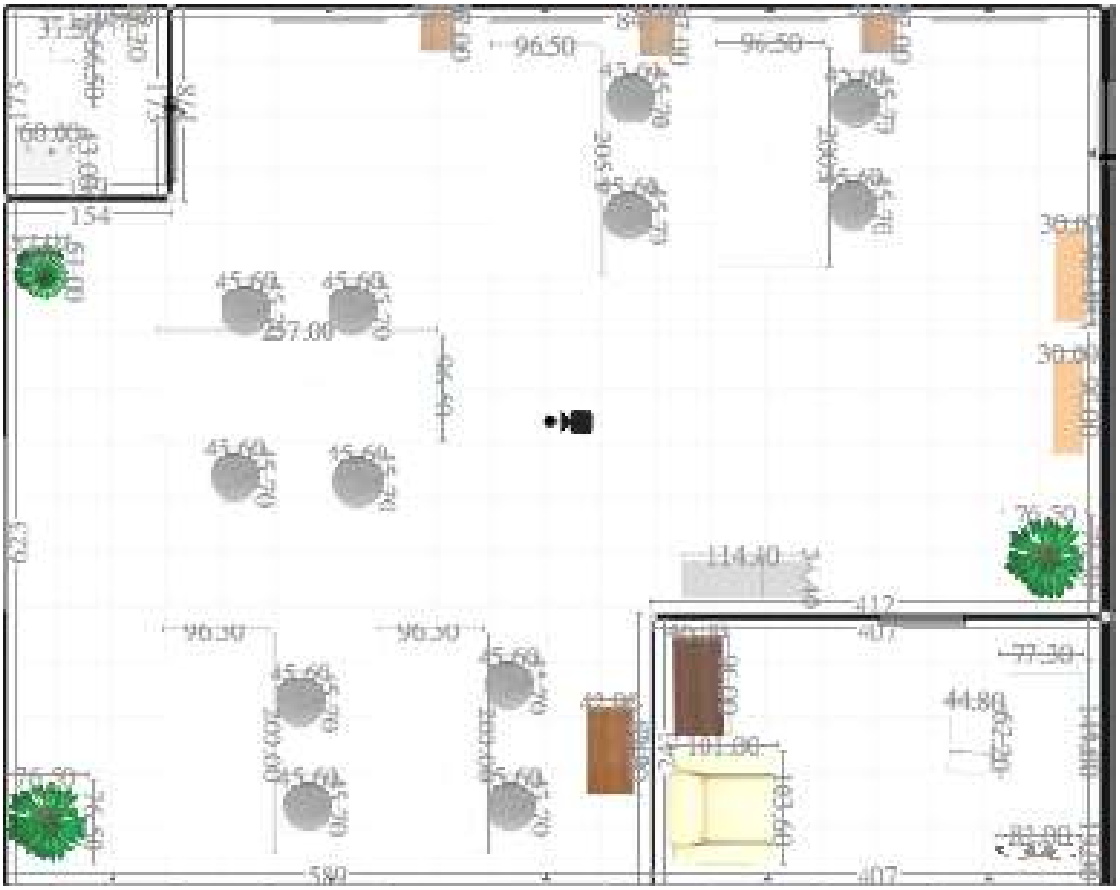


Figure 4.2:Plan 2D de l'espace de déploiement

4.10.2 Les résultats obtenus

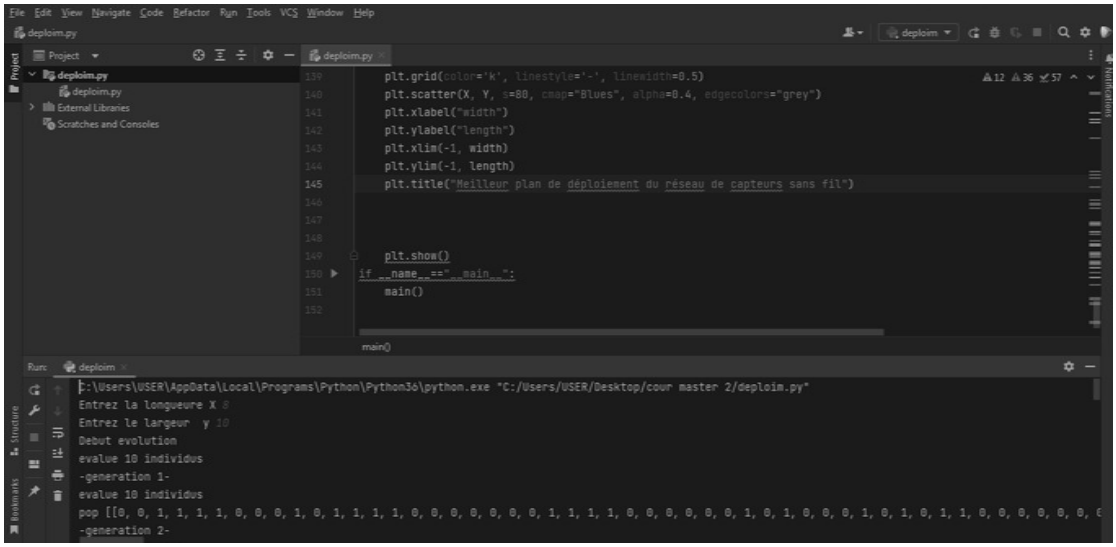


Figure (4. 3):Démarrage de l'évolution

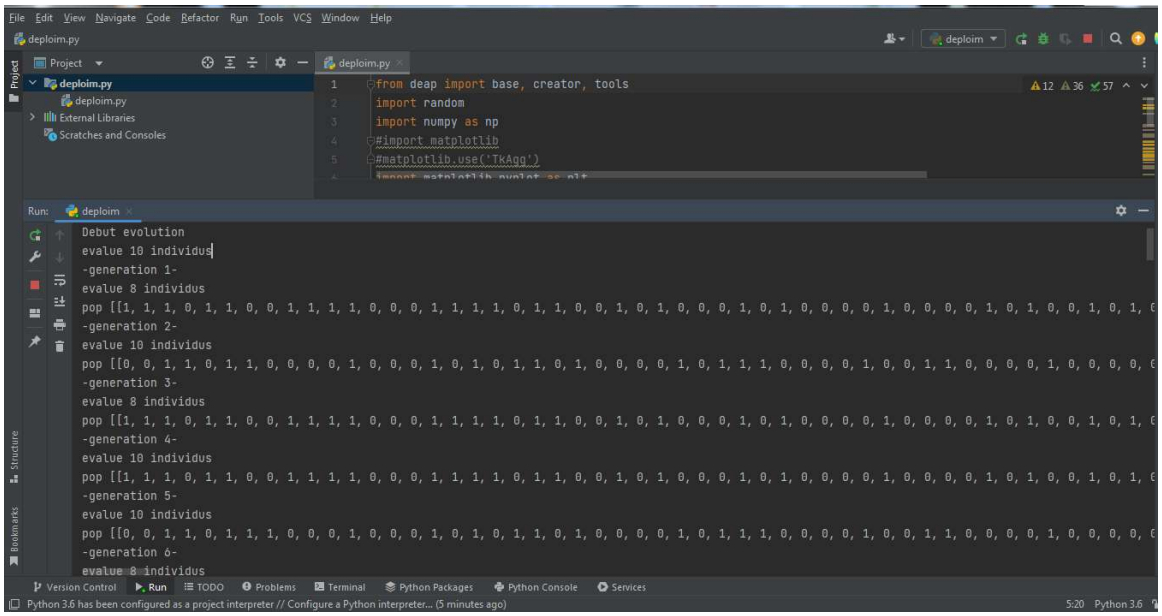


Figure 4.4: Itération des générations

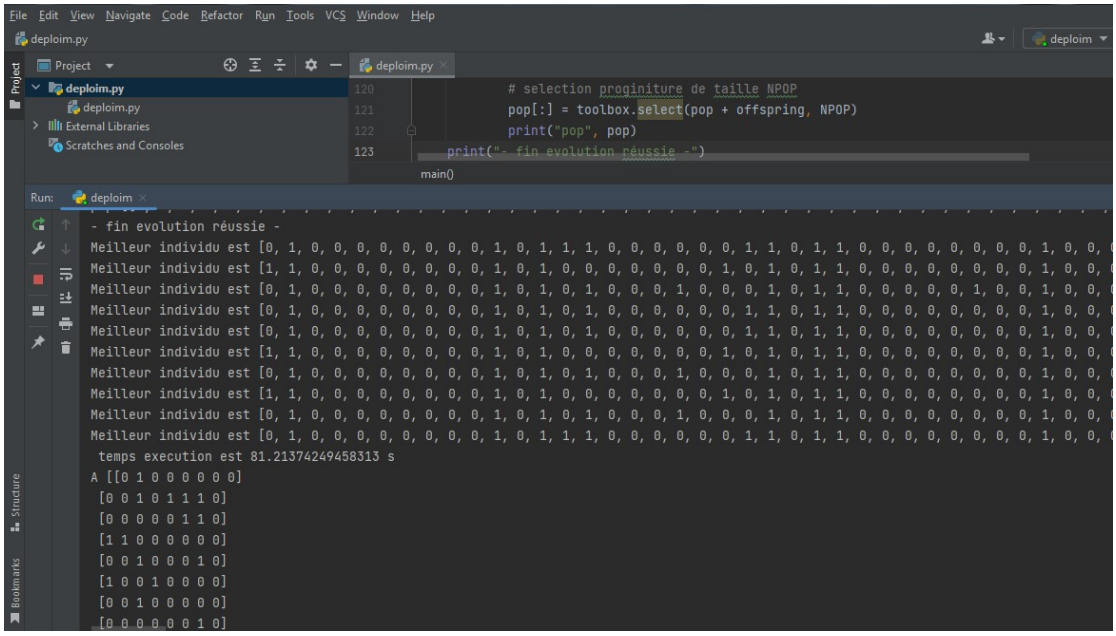


Figure 4.5:La meilleure solution de déploiement

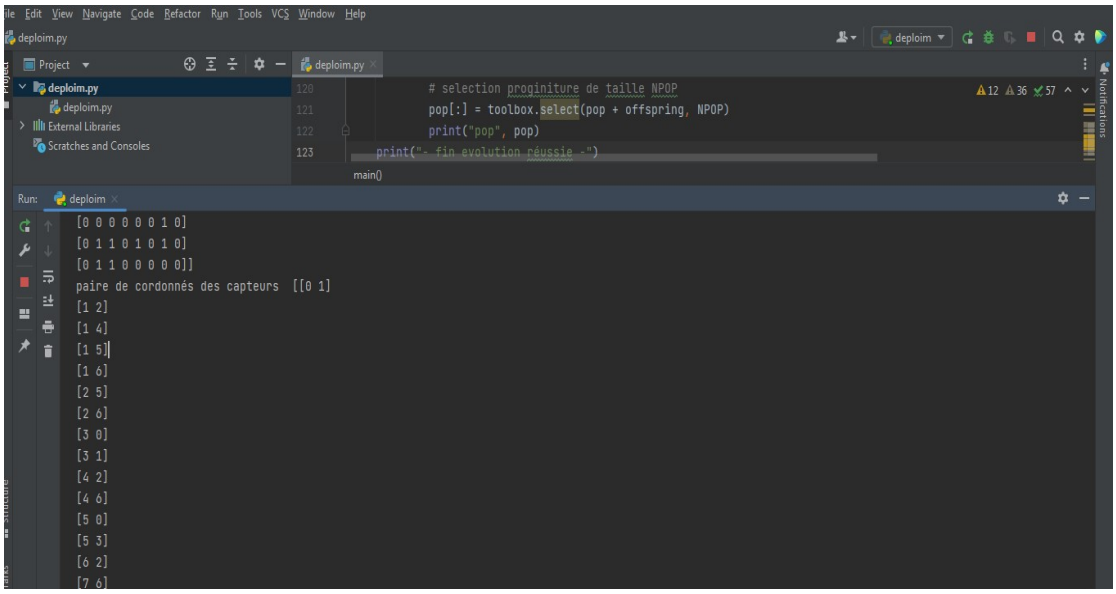


Figure 4.6 : les coordonnés physiques des capteurs

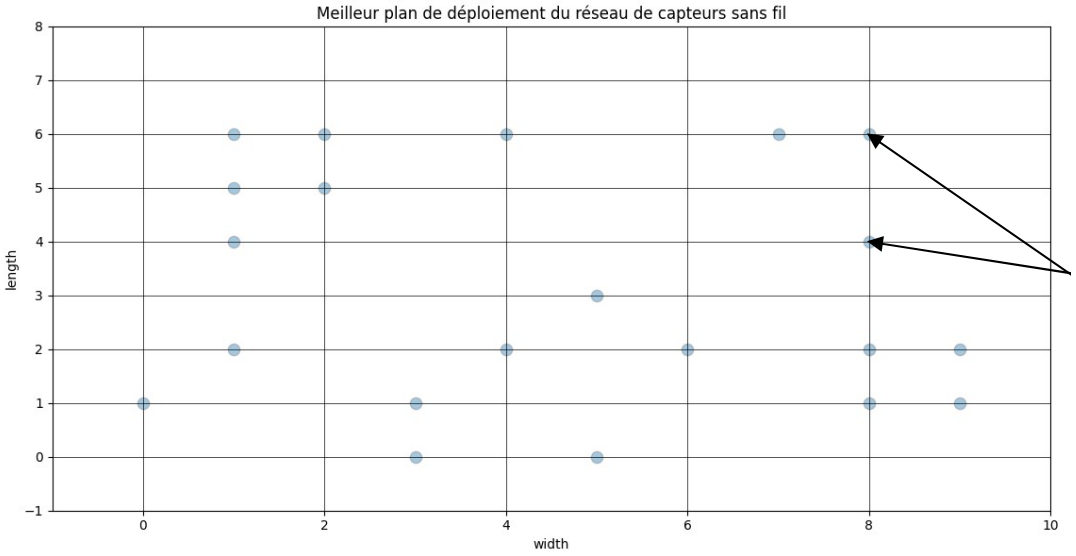


Figure 4.7:Un meilleur plan de déploiement du capteur de réseau sans fil

4.11 Conclusion

Dans ce dernier chapitre, nous avons d'abord présenté les outils et logiciels utilisés pour la résolution de notre problème. Ensuite, nous avons présenté le méthodes d'optimisation multi-objectif sélectionnée dans le chapitre précédent. Finalement, nous avons expliqué notre démarche pour la construction de l'algorithme.

Afin d'évaluer les performances de la méthode, nous avons considéré un cas d'étude. En premier lieu, nous avons défini la meilleure combinaison de paramètre à utiliser. Par la suite, nous avons lancé les simulations pour trouver la solution à notre problème. Les résultats obtenus ont montré que la méthode NSGA-II était plus adaptée à notre cas d'étude.

A la fin, nous retenons de notre travail que les algorithmes génétiques permettent de résoudre des problèmes complexes en un temps réduit et offrent à l'utilisateur une multitude de solutions exploitables. Il peut donc choisir la solution qui lui convient selon la configuration de son espace de déploiement et l'ordre d'importance qu'il donne aux objectifs. Leur inconvénient est qu'ils ont plusieurs paramètres qu'il faut gérer. Il est nécessaire de faire plusieurs essaies avant de pouvoir trouver la bonne combinaison de paramètres qui donne de meilleures performances aux algorithmes.

Conclusion générale

Ce projet de fin d'étude avait pour ambition de trouver une méthode de résolution du problème de déploiement d'un RCSF. Le but était d'optimiser les différentes métriques qui ont une influence sur les performances du réseau, à savoir le coût, la connectivité, la couverture et la sur-couverture. Une application à été développée afin de faciliter cette tâche. Cette application permet de trouver une solution qui répond aux besoins des utilisateurs selon les objectifs fixés.

Le problème que nous avons traité est considéré comme étant un problème d'optimisation multi-objectifs .Nous avons donc choisi la méthode la plus adaptée à ce type de problèmes les algorithmes génétique NSGA II Les résultats obtenus nous ont permis d'identifier la meilleure méthode à utiliser. Grace à cette méthode, nous avons pu établir un schéma de déploiement dans un espace type.

Une perspective intéressante serait l'utilisation d'autre méthode de d'optimisations multi-objectif comme SPEA II, NSGA III pour comparer et choisir la meilleure d'entre elles.

Bibliographie

[1] Soumaya 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.

[2] Pragati Kapil and Shashi Lata. Review on selecting topologies in zigbee networks.2016.

[3] G. Kaur and R. M. Garg. Energy efficient topologies for wireless sensor networks. International Journal of Distributed and Parallel systems (IJDPS), 3(5) :179-192,9 2012.

[4] Abderrahmen Belfkih, Bruno Sadeg, Claude Duvallet, and Laurent Amanton. Les bases de données dans les réseaux de capteurs sans fil. Techniques et sciences informatiques, 33 :739-776, 12 2014.

[5] Mohammad Matin. Wireless Sensor Network : Technology and Protocols. INTECH,09 2012.

[6] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm : Nsga-ii. IEEE transactions on evolutionary computation, 6(2) :182-197, 2002.

[7] Mohamed Amin Benatia. Multi-objective optimization of a network infrastructure dedicated to smart buildings. Thèses, INSA de Rouen, Decembre 2016.

[8] Siwiak Kazimierz. Ultra-Wideband Radio. John Wiley Sons, Ltd, 2005.

[9] Jaap C. Haartsen. The bluetooth radio system. IEEE Personal Communications, 7 :28-36, 2000.

[10] Hunn Nick. An introduction to wibree. White paper, 2006

[11] M. Farsi, M. A. Elhosseini, M. Badawy, H. Arafat Ali, and H. Zain Eldin. Deployment techniques in wireless sensor networks, coverage and connectivity : A survey. *IEEE Access*, 7 : 28940-28954, 2019.

[12] Harsimran Kaur and Rohit Bajaj. Review on localization techniques in wireless sensor networks. International Journal of Computer Applications, 116 :4-7, 04 2015.

[13] Subash Harizan and Pratyay Kuila. Evolutionary Algorithms for Coverage and Connectivity Problems in Wireless Sensor Networks : A Study, pages 257-280. Springer Singapore, Singapore, 2020.

[14] Zhibo Wang. Barrier Coverage in Wireless Sensor Networks. Theses, University of Tennessee, 2014.

[15] J. Amutha, Sandeep Sharma, and Jaiprakash Nagar. Wsn strategies based on sensors, deployment, sensing models, coverage and energy efficiency : Review, approaches and open issues. Wireless Personal Communications, 111(2) :1089-1115, 2020.

[16] Yi Zou and Krishnendu Chakrabarty. Sensor deployment and target localization based on virtual forces. volume 2, pages 1293 - 1303 vol.2, 03 2003.

[17] Pascal Rebreyend. “Algorithmes génétiques hybrides en optimisation combinatoire”. Thèse de Doctorat, Ecole Normale Supérieure de Lyon, 1999.

[18] M. Lott and I. Forkel. A multi-wall-and-floor model for indoor radio propagation. In IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No.01CH37202), volume 1, pages 464-468 vol.1, 2001.

[19] Charles Darwin. On the origin of species by means of natural selection. 1859. London : Murray Google Scholar, 1968.

[20] Ingo Rechenberg. Cybernetic solution path of an experimental problem. Royal Aircraft Establishment Library Translation 1122, 1965.

[21] Lawrence J Fogel, Alvin J Owens, and Michael J Walsh. Artificial intelligence through simulated evolution. 1966.

[22] John Holland. Adaptation in natural and artificial systems : an introductory analysis with application to biology. Control and artificial intelligence, 1975.

[23] John R Koza. Genetic programming : A paradigm for genetically breeding popu-lations of computer programs to solve problems, volume 34. Stanford University, Department of Computer Science Stanford, CA, 1990.

[24] David E.Goldberg. “Algorithmes génétiques Exploitation, optimisation et apprentissage automatique”. Addison-Wesley, France, SA, 1994.

[25] Blaise Madeline. “Algorithmes évolutionnaires et résolution de problèmes de satisfaction de contraintes en domaines finis”. Thèse de Doctorat en ciences, Université de Nice-Sophia antipolis, 2002.

[26] Youssef Harrat. “Contribution à l’ordonnancement conjoint de la production et de la maintenance : Application au cas d’un job Shop”. Thèse de Doctorat, L’U.F.R des Sciences et Techniques, Université de Franche-comté, 2003.

[27] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP : Evolutionary algorithms made easy. Journal of Machine Learning Research, 13 :2171-175, jul 2012.

[28] Nour El-Houda Benalia, Imene Si Hadj Mohand, Soumaya Ferhattaleb, Rabab Sadoun1 , Ahlem Bentrah, “MoEA-DeployWSN-SB: three variants of multi-objective evolutionary algorithms for the deployment optimization strategy of a WSN in a smart building”, Bharati Vidyapeeth’s Institute of Computer Applications and Management 2021

[29] Michèle Sebag et Marc Schoenauer. “Contrôle d’un algorithme génétique”. Revue d’intelligence artificielle Vol.10/1996.

[30] Jean-Marc Alliot, Thomas Schiex, Pascal Brisset, Frédérick Garcia. “Intelligence Artificielle et informatique théorique”. Paris, Cépaduès. 2ème Edition, 2002.

[31] Hanaa ZainEldin, Mahmoud Badawy, Mostafa Elhosseini, Hesham Arafat, and Ajith Abraham. An improved dynamic deployment technique based-on genetic algorithm (iddt-ga) for maximizing coverage in wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing,* pages 1-18, 2020.

[32] Yinggao Yue, Li Cao, and Zhongqiang Luo. Hybrid artificial bee colony algorithm for improving the coverage and connectivity of wireless sensor networks. *Wireless Personal Communications*, 108(3) :1719-1732, 2019

[33] Ali Afghantoloee and Mir Abolfazl Mostafavi. Towards optimal deployment of a sensor network in a 3d indoor environment for the mobility of people with disabilities (short paper). In 10th International Conference on Geographic Infor-mation Science (GIScience 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik,2018.

[34] Hanh Nguyen, Nam Nguyen Hai, and Huynh Binh. Particle swarm ptimization algorithms for maximizing area coverage in wireless sensor etworks. pages 893-904, 09 2018.

[35] Mohamed Amin Benatia, M'hammed Sahnoun, David Baudry, Anne Louis, Abdelkhalak El-Hami, and Belahcene Mazari. Multi-objective wsn deployment using genetic algorithms under cost, coverage, and connectivity constraints. Wireless Personal Communications, 94(4) :2739-2768, 2017.

[36] Jingwen Tian, Meijuan Gao, and Guangshuang Ge. Wireless sensor network node optimal coverage based on improved genetic algorithm and

binary ant colony algorithm. EURASIP Journal on Wireless Communications and Networking, 2016(1) :1-11, 2016.

[37] Vahab Akbarzadeh, Julien-Charles Lévesque, Christian Gagné, and Marc Parizeau. Efficient sensor placement optimization using gradient descent and probabilistic coverage. *Sensors (Basel, Switzerland)* 14 :15525-52, 08 2014.

[38] Nadjib Aitsaadi, Nadjib Achir, Khaled Boussetta, and Guy Pujolle. Artificial potential _eld approach in wsn deployment : Cost, qom, connectivity, and lifetime constraints. Computer Networks, 55(1) :84-105, 2011.

[39] Jet Brains. Pycharm : L'IDE python pour développeurs professionnels, 2020.