

10/2015 - M/INF

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI
BOUMEDIENE (U.S.T.H.B)
Faculté d'Electronique et d'Informatique



MEMOIRE

Présenté pour l'obtention du diplôme de **MAGISTER**
En **INFORMATIQUE**
Spécialité : Systèmes Intelligents et Ingénierie du logiciel
par : Mr **BENMEZAL Larbi**

Sujet

**Une approche à voisinage pour le
Problème de Positionnement des
Antennes en réseaux GSM**

Soutenu publiquement le 23/02/2015 devant le jury :

<i>M Karim ATIF</i>	<i>Maître de conférence/A à l'USTHB</i>	<i>Président</i>
<i>Mlle Dalila BOUGHACI</i>	<i>Maître de conférence/A à l'USTHB</i>	<i>Directrice de mémoire</i>
<i>M Mourad DAOUDI</i>	<i>Maître de conférence/A à l'USTHB</i>	<i>Examineur</i>
<i>M Hakim AITZAI</i>	<i>Maître de conférence/A à l'USTHB</i>	<i>Examineur</i>

REMERCIEMENTS

Avant toute chose je commence par louer Dieu qui par son bienfait et ça grâce m'a permis d'accomplir ce travail.

*J'adresse mes remerciements aux personnes qui m'ont aidé dans la réalisation de ce mémoire, en particulier **Mlle Boughaci Dalila** ma directrice de recherche premièrement pour accepter d'être le guide qui m'a aidé dans mes premiers pas dans la recherche scientifique, deuxièmement pour son aide et conseils qui ont permis de mener à bien ce travail.*

*Je remercie vivement **M Karim Atif** pour l'honneur qu'il m'a fait en acceptant de présider le jury de ma soutenance.*

*Mes remerciements vont aussi à **M Hakim Aitzai** et **M Mourad Daoudi** d'avoir accepté de faire partie du jury.*

Dédicaces

Je dédie ce modeste travail à mes parents qui ont œuvré pour ma réussite, de par leur amour, leur soutien, tous les sacrifices consentis et leurs précieux conseils. Sans eux ce travail n'aurait jamais pu être réalisé. Recevez à travers ce travail, l'expression de mes sentiments et de mon éternelle gratitude.

A ma grande mère qui était mon soutien moral et qui voulait tellement me voir réussir, je te dédie ce travail aussi modeste soit-il, qu'Allah te garde dans son vaste paradis.

Aux personnes dont j'ai bien aimé la présence dans ce jour, à mes sœurs et membres de la famille je dédie ce travail.

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, mes aimables amis et collègues d'étude.

Larbi.

Résumé

Les métas heuristiques sont définis comme des processus de génération itératifs qui guident une heuristique subordonnée en combinant intelligemment différents concepts pour l'exploration et l'exploitation de l'espace de recherche. Il existe plusieurs types de métas heuristiques, parmi eux les métas heuristiques évolutionnaire qui traitent un ensemble de solutions appelées population, et les métas heuristiques de voisinage qui traitent une solution à la fois et déterminent son voisinage pour passer à une nouvelle solution. Dans ce mémoire nous nous sommes intéressés au deuxième type de métas heuristiques, à savoir les métas heuristiques de voisinage.

Parmi les algorithmes de voisinage qui sont connus pour avoir donnés de bons résultats dans différents problèmes d'optimisation sont les algorithmes ILS (Iterated Local Search) et BLS (Breakout Local Search) qui est inspiré d'ILS. Nous avons appliqué ces algorithmes au problème de positionnement des antennes en réseaux GSM. Nous avons aussi proposé un algorithme qui est aussi inspiré de ILS.

Le problème APP (Antenna Positioning Problem) englobe un ensemble de décisions concernant le déploiement des ressources physiques dans les réseaux GSM. Essayant d'optimiser leurs utilisations (minimiser le nombre d'antennes utilisés) tout en optimisant les performances du réseau (maximiser la couverture réseau, minimiser le bruit...) en prenant en compte certaines contraintes qui doivent être respectées dans les réseaux GSM. Le problème peut s'étendre aussi à spécifier le type et la configuration d'antennes qui permettent d'atteindre les objectifs. Par contre, le problème APP n'est généralement pas considéré avec toutes les contraintes est objectifs lors de sa résolution en raison de diminuer sa complexité. Dans notre travail nous avons considéré comme objectif de maximiser le taux de couverture de la zone géographique et en même temps minimiser le nombre d'antennes utilisées.

Plusieurs expérimentations ont été réalisées sur des benchmarks dont un représente des données réelles et deux sont générés aléatoirement. Les résultats ont prouvé l'efficacité des deux algorithmes mais surtout de notre algorithme proposé.

Mots clés : Problème de positionnement d'antennes en réseaux GSM, Métaheuristiques, Recherche locale, Optimisation combinatoire.

Sommaire

1	Introduction générale.....	6
	Chapitre 1 :	9
	Définition du problème de placement d'antennes et modélisation.	9
1	Introduction	10
2	Notions de bases sur les réseaux GSM.....	10
2.1	Composants de base des réseaux GSM.....	10
2.1.1	Les stations de base (BTS) :	10
2.1.2	Les contrôleurs de stations de base (BSC) :	10
2.1.3	Les centres de commutations mobiles (MSC) :.....	10
2.1.4	Les registres de localisation des visiteurs (VLR) :	11
2.1.5	L'enregistreur de localisation nominale (HLR).....	11
2.1.6	Antennes	12
2.2	Configurations d'antennes.....	13
2.3	Multiplexage fréquentiel et temporel	14
2.4	Cellules et réutilisation de fréquences.....	14
2.5	Interférences	15
2.6	Le rapport C/I.....	15
3	Problème de positionnement d'antennes (APP)	16
3.1	Définition du problème APP	17
3.2	Modélisation du problème.....	17
3.2.1	Modélisation de Reininger.....	17
3.2.2	Modélisation de Calegari et al	22
3.2.3	Modélisation de Mendes.....	23
3.3	Modèles de propagations d'ondes	25
3.4	Complexité du problème	25

4	Conclusion	26
	Chapitre 2 :	27
	Méthodes de résolution approchées (Approches évolutionnaires et approches de voisinage).....	27
1	Introduction	28
2	Travaux antérieurs	28
3	APP comme problème d'optimisation.....	29
3.1	Espace de recherche (de décision) :	29
3.2	Fonction objectif :	30
3.3	Contraintes:	30
3.4	Codage de solution	30
4	Optimisation multiobjectif.....	32
5	Méthodes d'optimisations (Méthodes de résolutions).....	34
5.1	Définition des métaheuristiques	35
5.2	Classification des métaheuristiques	35
6	Approche mono objectif	36
6.1	Métaheuristiques évolutionnaires (EA).....	36
6.1.1	Les algorithmes génétiques	37
6.1.2	Principaux étapes d'un algorithme génétique.....	38
6.1.3	Algorithmes génétique générationnel et stationnaire	39
6.1.4	Le problème APP résolu avec ssGA et gGa	39
6.1.5	Algorithme CHC.....	39
6.1.6	APP résolu avec CHC.....	41
6.1.7	Algorithme PBIL	42
6.2	Méthodes basées sur la recherche locale.....	45
6.2.1	La recherche locale	45
6.2.2	Algorithme Simulated Annealing (SA)	46

6.2.3	La recherche Taboue	49
6.3	Algorithme de recherche ILS (Iterated Local Search)	51
6.3.1	Génération de la solution initiale.....	52
6.3.2	Mécanisme de perturbation	52
6.3.3	Mécanisme d'intensification.....	52
7	Approches multiobjectif	52
7.1	SSGA.....	53
7.1.1	Partage et élitisme.....	54
7.1.2	Prise en compte des contraintes	56
7.2	Algorithme NSGA.....	56
7.3	NSGA-II.....	58
7.3.1	Distance de Crowding	60
7.4	SPEA 2.....	62
7.4.1	Affectation de performance	63
7.5	Algorithmes MOCHC	65
7.5.1	Prise en charge des contraintes	65
7.5.2	Performances MOCHC.....	65
8	Approches parallèles.....	66
8.1	Modèle proposé par Caligari	66
8.2	Autres travaux sur les approches parallèles	68
9	Conclusion.....	69
Chapitre 3 :		70
Approche proposée		70
1	Introduction	71
2	Délimitation du problème résolu	71
3	Modélisation	71

3.1	La zone géographique	71
3.1.1	Matrice.....	72
3.1.2	Point.....	72
3.2	Antennes.....	72
4	Fonction objectif utilisée	73
5	Evaluation d'une solution.....	73
5.1	Algorithme de couverture.....	73
5.2	Algorithme d'évaluation de la solution.....	74
5.3	Voisinage.....	75
6	Algorithme ILS.....	76
6.1	Mécanisme de perturbation d'ILS.....	77
6.2	Recherche local employée par ILS.....	77
6.3	Algorithme de perturbation employé par ILS	78
6.4	Critère d'acceptation	78
7	Algorithme BLS	79
7.1	Fonctionnement de BLS.....	79
7.1.1	La recherche locale dans BLS	79
7.1.2	Mécanisme de perturbation de BLS	80
8	Notre approche	86
9	Phases de résolution.....	89
9.1	Phase de prétraitement	89
9.2	Phase de traitement.....	91
9.3	Phase de post-traitement.....	91
10	Conclusion.....	92
	Chapitre 4 :	93
	Résultats expérimentaux.....	93

1	Introduction	94
2	Benchmark.....	94
3	Paramètres	95
4	Critères de comparaison entre algorithmes	95
4.1	La valeur de la fonction objectif	96
4.2	Nombre d'évaluations	96
5	Résultats.....	96
5.1	Résultats obtenus sur la première instance.....	98
5.2	Résultats obtenus sur la deuxième instance	102
5.3	Résultats sur la troisième instance	105
6	Comparaison avec les résultats obtenus dans le travail de Mendes et al.	109
7	Conclusion	110
	Conclusion générale	111

1 Introduction générale

La téléphonie mobile est devenue de nos jours une technologie à la portée de tout le monde. Alors que des années avant ce n'était pas le cas. Ce phénomène indique qu'une grande évolution a eu lieu dans ce domaine. En effet, avec la croissance des services utilisateurs, beaucoup de domaines d'études ont apparus, et beaucoup d'entre eux sont des problèmes d'optimisation. La planification des réseaux GSM en est un exemple.

La planification d'un réseau GSM comporte principalement deux problèmes : Le problème de positionnement d'antennes (APP) et l'allocation de fréquences (FAP). Le problème APP peut être résumé par le choix des emplacements (sites) pour positionner les antennes, le nombre d'antennes à positionner dans chaque site ainsi que le paramétrage de ces antennes. Par contre le problème FAP consiste à assigner aux antennes du réseau des fréquences parmi un ensemble de fréquences disponibles. Ces problèmes sont tous deux des problèmes NP-difficile. Le problème d'optimisation APP est l'objet de notre étude dans ce mémoire.

Plusieurs méthodes de résolutions existent pour des problèmes de ce type c.à.d. des problèmes d'une importante complexité. Les méthodes de résolutions approchées sont généralement les méthodes les mieux adaptées pour la résolution de ces problèmes, surtout lorsque le but souhaité est d'obtention d'une bonne solution dans des délais raisonnables au lieu de chercher la meilleure solution dans un délai important.

Après avoir fait une étude de plusieurs métaheuristiques dans l'état de l'art, nous nous sommes intéressés aux métaheuristiques de voisinage plus précisément les algorithmes ILS et BLS. Une approche de voisinage a été aussi proposée afin de résoudre le problème APP.

Ce mémoire a été organisé de la manière suivante : Dans le premier chapitre nous présentons quelques notions sur les réseaux GSM pour la bonne compréhension du problème de positionnement d'antennes, ainsi que quelques approches qui ont été adoptées pour le modéliser. Dans le deuxième chapitre, nous présentons les méthodes de résolutions, à savoir les métaheuristiques évolutionnaires et les métaheuristiques de recherche locale. Dans le troisième chapitre nous présentons les trois approches :ILS,

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

BLS ainsi que notre approche. Le dernier chapitre est consacré aux résultats expérimentaux. Enfin, nous terminons avec une conclusion et perspectives.

Chapitre 1 :

Définition du problème de placement d'antennes et
modélisation.

1 Introduction

Dans ce chapitre, nous allons faire le point sur les différentes caractéristiques et notions techniques des réseaux GSM afin de mieux appréhender le problème de placement d'antennes. Ensuite, on va présenter quelques modélisations de ce dernier.

2 Notions de bases sur les réseaux GSM

2.1 Composants de base des réseaux GSM

Un réseau GSM est constitué de plusieurs composantes dont nous citons les stations de base, les contrôleurs de stations de base et les centres de commutation mobile. Ces principales composantes sont détaillées dans ce qui suit afin de bien illustrer le rôle de chacun d'eux.

2.1.1 Les stations de base (BTS) :

C'est l'interface entre le téléphone mobile et le reste du réseau. Une station de base contient tous les émetteurs-récepteurs appelés TRX reliés à la cellule et dont la fonction est de transmettre et recevoir des informations sur le canal radio en proposant une interface physique entre la station mobile et le contrôleur de station de base BSC.

2.1.2 Les contrôleurs de stations de base (BSC) :

Ils gèrent la coordination entre les relais radio (BTS).

- La gestion et la configuration du canal radio : il doit choisir pour chaque appel la cellule la mieux adaptée et doit sélectionner à l'intérieur de celle-ci le canal radio le plus adapté à la mise en route de la communication.
- La gestion du *handover* (prise de main) intra BSC : il décide, sur la base des relevés reçus par la BTS, le moment pour effectuer le *handover*, autrement dit, le changement de cellule lors des déplacements de l'utilisateur pendant une conversation.
- Les fonctions de décodage des canaux radio Full Rate (16 kbps) ou Half Rate (8 kbps) pour des canaux à 64 kbps.

2.1.3 Les centres de commutations mobiles (MSC) :

Ces entités sont responsables de l'acheminement des communications dans le réseau et assurent également l'interconnexion entre le réseau de téléphone cellulaire et

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

le réseau fixe traditionnel. Elles génèrent toutes les informations de taxation et gèrent la complexité des connexions due aux déplacements pendant la communication.

2.1.4 Les registres de localisation des visiteurs (VLR) :

Le VLR est une base de données reliée à un MSC qui stocke temporairement les informations qui concernent chaque mobile dans la zone de travail du MSC, (identité de l'abonné, sa dernière zone de localisation, les services complémentaires souscrits par celui-ci, les éventuelles restrictions ou interdictions d'établissement de la communication).

2.1.5 L'enregistreur de localisation nominale (HLR)

Le HLR est une base de données qui contient des informations essentielles pour les services de téléphonie mobile et avec un accès rapide de manière à garantir un temps d'établissement de connexion aussi court que possible. Elle contient toutes les informations relatives aux abonnés : le type d'abonnement, la clé d'authentification (relatif à la carte SIM), les services souscrits, le numéro de l'abonné, etc. HLR contient aussi certain nombre de données dynamiques telles que la position de l'abonné dans le réseau et l'état de son terminal (allumé, éteint, en communication, libre, ...).

Les stations de base et leurs contrôleurs sont les constituants du sous système radio BSS *Base Station Sub-system* (le système qui gère la transmission radio).

Le MSC, HLR, VLR et l'Auc (le centre d'authentification) sont les composants du sous système réseau NSS (*Network Station Sub-system*). Ce dernier assure principalement les fonctions de commutation et de routage.

Cette infra structure est gérée par le centre d'exploitation qui permet la gestion et la supervision du réseau (Le sous-système opérationnel OSS).

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

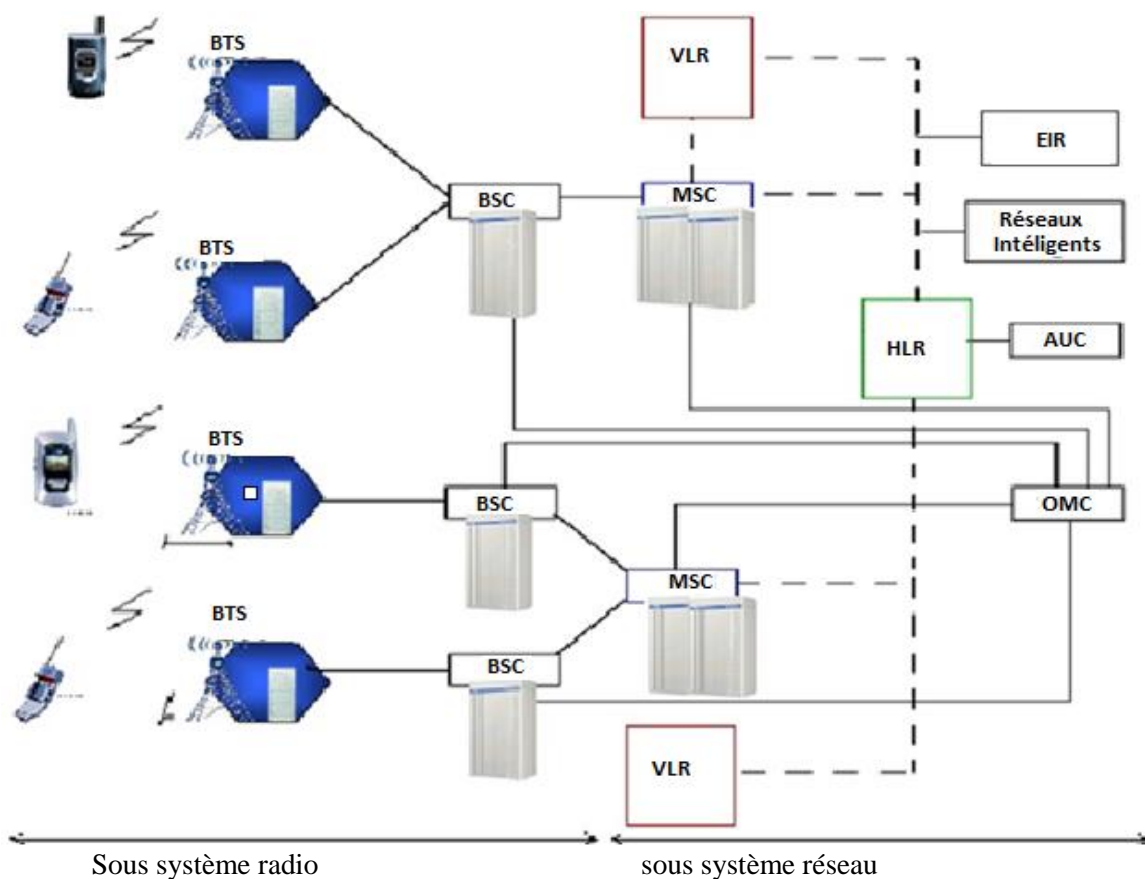


Figure 1.1: infrastructure d'un réseau GSM

2.1.6 Antennes

Une antenne est l'unité apparente du réseau GSM. En générale, un réseau comporte plusieurs types d'antennes qui sont caractérisées par leurs gains de transmission (gain d'antenne) et leurs diagrammes de rayonnement.

Généralement, Les travaux qui ont considérés le type d'antennes dans la résolution du problème APP [37] [16] ont choisi 3 types d'antennes : *omnidirectional* (OMNI), *large directional* (LD) et *small directional* (SD). Dans le cas où des antennes directionnels sont utilisés, les relais sont souvent composés de trois antennes-panneaux orientées à environ 120° l'une de l'autre, de manière à couvrir sur 360° . En fait, ce dernier type est souvent le plus utilisé.

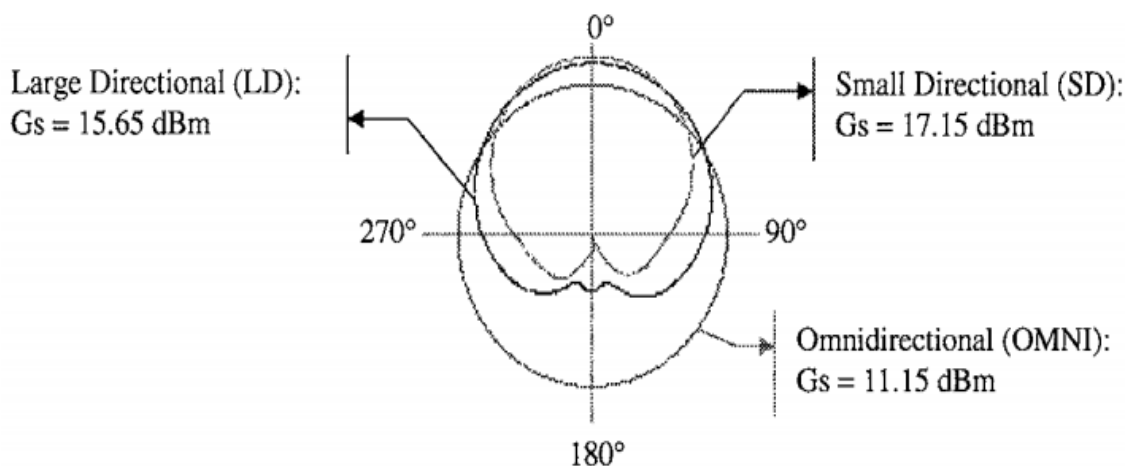


Figure 1.2: Forme de cellules générées par des antennes LD, SD et OMNI.

2.2 Configurations d'antennes

a. Tilt

Le tilt d'une antenne correspond à son angle d'inclinaison dans un plan vertical. En "tiltant" l'antenne vers le bas (ce qui correspond à une augmentation de l'angle du tilt), la zone couverte par l'antenne diminue et l'intensité de la puissance moyenne reçue dans la cellule augmente.

b. Azimut

L'angle d'azimut correspond à l'orientation du lobe principal de l'antenne dans le plan horizontal. Idéalement, dans un site tri-sectoriel, l'orientation des antennes est séparée de $2\pi/3$. Une modification de l'angle d'azimut peut être utile suite à un effet de masque lié au paysage ou à des bâtiments, qui peuvent causer des réflexions et des interférences indésirables.

c. Gain - Puissance

Signifie l'amplification que l'antenne effectue du signal d'entrée et s'exprime en dBi (décibel). Les antennes omnidirectionnelles ont une puissance qui varie entre 2 à 11 dBi et une puissance qui va jusqu'à 18 dBi pour les antennes directionnelles. La puissance émise appelée PIRE (Puissance Isotrope Rayonnée Équivalente) est fournie par la BTS. Référée aussi par PAR (Puissance Apparente Rayonnée, $PAR = PIRE - 2,15 \text{ dB}$).

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

2.3 Multiplexage fréquentiel et temporel

La norme GSM utilise deux bandes de fréquences 890 - 915 [MHz] pour les communications montantes (du mobile vers la station de base) et 935 - 960 [MHz] pour les communications descendantes. Pour une communication, un canal de 200 [kHz] est utilisé, ce qui laisse 124 canaux possibles. Connaissant les différents canaux disponibles, il est possible d'effectuer un multiplexage fréquentiel pour obtenir des fréquences porteuses à attribuer aux émetteurs/récepteurs des stations de base.

Les canaux obtenus à partir du multiplexage fréquentiel (FDMA) sont subdivisés en 8 intervalles de temps, afin de permettre par exemple de faire parler huit utilisateurs l'un après l'autre dans le même canal. C'est ce qu'on appelle le multiplexage temporel (TDMA).

2.4 Cellules et réutilisation de fréquences

Un concept très important dans les réseaux GSM est celui de la cellule. En effet, afin de bien prendre en charge le trafic du réseau, ce dernier est décomposé en cellules. Les premières générations des réseaux GSM, mettaient en œuvre des cellules de grandes tailles (environ un rayon de 50 Km), ce qui n'offrait pas de bonne performances (puisque le nombre simultané d'utilisateur du réseau ne pouvant pas dépasser le nombre de bandes de fréquences disponibles). Ainsi le concept de cellule qui signifie la décomposition du territoire en plus petite zones permet de réduire le nombre de personnes pouvant être pris en charge par une station.

Cette subdivision du réseau nous mène à un concept très important qui est le principe de réutilisation de fréquences. En effet, les cellules étant plus petites, l'opérateur réseau peut manager les canaux de fréquences disponibles de telle sorte que des fréquences similaires peuvent être utilisées dans plusieurs cellules. L'allocation de fréquences est soumise à la condition qui exige que deux cellules adjacentes ne doivent pas avoir la même fréquence. Et la distance minimale entre deux cellules utilisant la même fréquence doit être supérieure à $D = R\sqrt{3N}$, où R est le rayon de la cellule, et N et le nombre de canaux disponibles.

Les zones de couvertures doivent se recouvrir d'environ 10% afin de garantir à un utilisateur passant d'une cellule à une autre que son appelle soit toujours pris en charge par la nouvelle cellule.

2.5 Interférences

Les fréquences de porteuses utilisées pour la transmission radioélectrique subissent plusieurs perturbations ou interférences qui sont d'origine d'autres signaux envoyés par les autres stations. On peut distinguer deux types d'interférences:

- **Co-canal** : dues aux signaux émis par les autres stations qui utilisent la même fréquence
- **Canaux adjacents** : dues aux signaux émis par les stations de base utilisant des fréquences voisines.

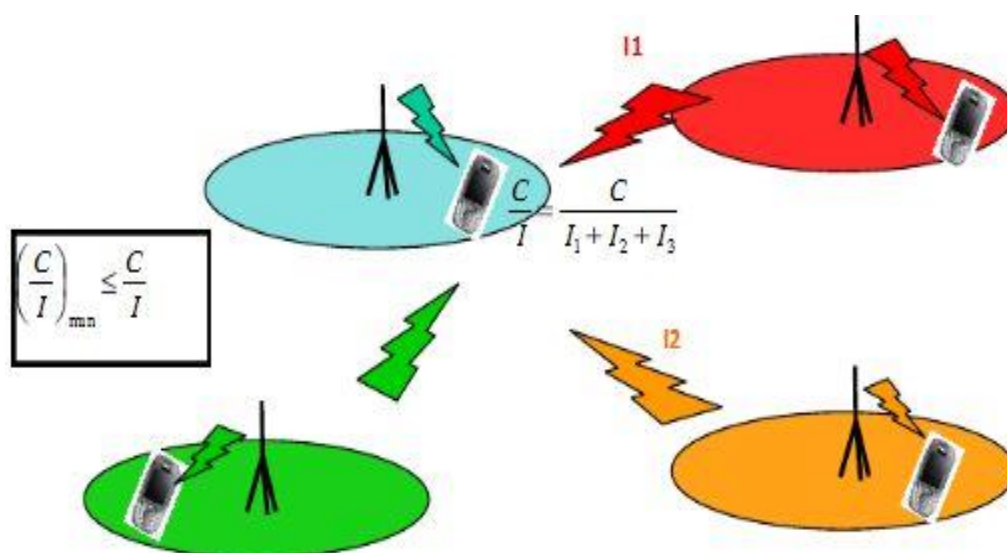


Figure1.3 : Illustration des interférences

C : La puissance du signal utile.

I : Le niveau d'interférence.

2.6 Le rapport C/I

Plusieurs techniques sont employées dans la norme GSM pour essayer de réduire l'impact des interférences. On cite par exemple :

- **Le saut de fréquence**: (*Frequency Hopping*): Se base sur une technique de changement de fréquence porteuse lors de la transmission. Ainsi si une bande de fréquence est exposée à des perturbations suffisantes pour perturber les données, seulement une petite quantité sera perdue.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

- **Le décalage temporel des envois** (*Timing Advance*): Les mobiles les plus distants parmi ceux qui envoient des informations sur des valeurs de porteuses proches doivent avancer le moment d'envoi afin d'éviter que leurs messages soient arrivés tard à la station de base, donc interférés avec les messages reçus du prochain slot.
- **Le codage**: différentes techniques de codages sont utilisées afin de minimiser l'impact des erreurs qui peuvent se produire lors des communications.

L'un des objectifs de l'APP est de maximiser le rapport C/I. c'est-à-dire, diminuer l'interférence par rapport à la puissance du signal. Ce qui permettra d'offrir une bonne qualité de service assurée par le réseau.

3 Problème de positionnement d'antennes (APP)

Plusieurs problèmes d'optimisations surviennent lors de la planification d'un réseau GSM. Généralement le but est de rationaliser l'utilisation des ressources, ie : maximiser les performances tout en minimisant le cout. Parmi ces problèmes on trouve le problème de positionnement d'antennes **APP** et le problème d'allocation de fréquences **FAP**. Le premier consiste à définir l'emplacement et la configuration d'antennes (Azimute, puissance, tilt...) de telle sorte à maximiser la couverture et à minimiser le coût (minimiser le nombre d'antennes). Le second consiste à attribuer à chaque antenne une fréquence afin de minimiser les interférences.

L'APP et le FAP sont dans certains cas étudiés ensemble comme dans [1], comme ils peuvent être étudiés séparément, résolvant premièrement l'APP ensuite le FAP séparément.

L'avantage de considérer les deux problèmes simultanément est que l'interaction entre eux est prise en compte. Donc de meilleurs résultats sont obtenus. Par contre, la complexité du problème est considérablement élevée. C'est pour palier à ce problème que d'autres préfèrent étudier les deux problématiques séparément, en ajoutant la contrainte de minimiser l'interférence dans l'étude de l'APP afin de faciliter plus tard l'étude du FAP.

Dans notre cas, nous n'allons pas aborder le deuxième problème (FAP). En effet, nous allons nous intéresser au problème de positionnement d'antennes.

3.1 Définition du problème APP

Le problème APP (Antenna Positioning Problem) englobe un ensemble de décisions concernant le déploiement des ressources physiques dans les réseaux GSM. Essayant d'optimiser leurs utilisations (minimiser le nombre d'antennes utilisés) tout en optimisant les performances offertes par le réseau (maximiser la couverture réseau, minimiser le bruit...) en prenant en compte certaines contraintes qui doivent être respectées dans les réseaux GSM. Le problème peut s'étendre aussi à spécifier le type et la configuration d'antennes qui permettent d'atteindre les objectifs.

Les éléments de base considérés généralement dans un problème APP sont : la surface de travail géographique où le signal radio et le trafic sont mesurés, les stations mobiles et les antennes. Donc le problème consiste à déterminer quel sont parmi un ensemble de sites prédéfinis les sites dont on va utiliser (installer) les antennes de tel sorte à satisfaire des objectifs et des contraintes définis à l'avance.

Le problème de positionnement d'antennes est un problème hautement combinatoire, NP-difficiles [25]. C'est-à-dire, la résolution se heurte à une explosion du nombre de combinaisons à explorer.

3.2 Modélisation du problème

Comme tout problème réel devant être résolu par une machine nécessite une modélisation, Le problème APP a aussi été abordé selon différentes modélisations, chacune possédant ses propres caractéristiques.

Nous présentons dans ce qui suit, quelques modélisations qui ont été les plus utilisées dans les approches de résolution de l'APP.

3.2.1 Modélisation de Reininger

Reininger a proposé dans [29] une formulation analytique du problème APP. Cette représentation a été un moyen utilisé par plusieurs travaux comme celui dans [26]. Voici quelques définitions issues de cette modélisation.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

3.2.1.1 Espace de travail (Working area)

L'espace de travail est la zone géographique à couvrir, modélisé par des points de teste et noté par P . Il est décrit par une base de données contenant 4 types de points :

Un ensemble L de sites candidats pour le positionnement des stations de bases notées ici (BS). $L = \{L_i / i \in \mathbb{N}\}$. où chaque site est défini par ces coordonnées (x,y) et des fois z (hauteur).

Un ensemble R de points de teste de réception (RTP) dans lesquels le signal radio va être testé, tel que $R = \{R_i / i \in \mathbb{N}\}$. Chaque RTP peut être utilisé comme un point de teste de signal pour calculer la couverture réseau.

Un ensemble ST de point de teste de service (STP) dans lesquels le service attendu est calculé, tel que $ST = \{ST_i / i \in \mathbb{N}\}$. ST définit l'ensemble des STP où le réseau doit surmonter un seuil de qualité du signal pour assurer une qualité de service donné (QoS). Ce seuil, S_q , dépend du type de mobile et la location. Les STP sont définis pour définir la notion de cellule. Un ensemble de points STP couverts par une BS représente la cellule associé à cette BS.

Un ensemble TT de points de teste du trafic (TTP) dans lesquels la quantité du trafic attendue va être testée, $TT = \{T_i / i \in \mathbb{N}\}$. Chaque TTP est associé avec la qualité du trafic dans ce point donnée en *Erlang* (unité de mesure du trafic). Le nombre de transmetteur (TRX) qu'une station de base va supporter dépend de la quantité globale du trafic dans ses TTP.

Chaque T_i est associé à un ST_i (ont les même coordonnées) pour assurer une bonne qualité du signal dans les points où le trafic est localisé. Et chaque point ST_i est associé à un point R_i afin de connaître le signal radio dans ce point. Donc, nous avons la formule suivante:

$$TT \subseteq ST \subseteq R \text{ (voir figure 1.4).}$$

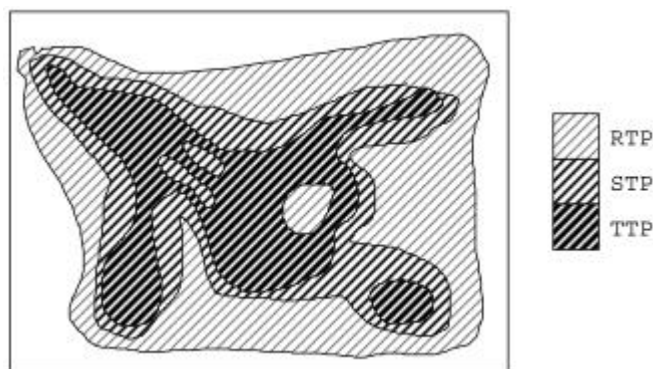


Figure 1.4 : la relation entre les ensembles TT , ST et R .

3.2.1.2 Cellule

Une cellule associée à une station de Base (BS) est l'ensemble des points STP ayant cette BS comme le meilleur serveur. On note une cellule associée à une station de base b par :

$$Cell(b) = \{ p \in ST / Cd_{b,p} \geq S_q \text{ et } \forall b' \in BS1 \quad b' \neq b \quad Cd_{b,p} > Cd_{b',p} \}$$

Où $BS1$ est l'ensemble des stations de bases actives (Sites candidats dont on a installé des stations de bases) et $Cd_{b,p}$ est la puissance du signal radio reçu par le point p (de type STP) de la part de la station de base b .

Un modèle de propagation d'ondes est généralement utilisé pour simuler la diffusion du signal afin de calculer sa puissance dans les points de la zone de travail.

3.2.1.3 Contraintes et objectifs

a. Mobilité (*handover*)

Toutes cellules doit posséder une zone dont l'intersection avec les cellules voisines est différente de l'ensemble vide (STPs communs), c'est-à-dire une zone qui est couverte par plus d'une station de base. Ceci pour assurer la mobilité (*handover*), lorsqu'une station mobile se déplace entre cellules, la communication sera donc garantie de toutes coupures.

Techniquement, il doit toujours y'avoir dans une cellule couverte par une BS b des points STP p dont au moins une autre BS b' tel que $Cd_{b',p}$ est supérieur au seuil S_q et inférieur à $Cd_{b,p}$ par au moins 7 dBm.

$$handCell(b) = \{ p \in Cell(b) / \exists b' \in BS1 \text{ et } Cd_{b',p} \geq S_q \text{ et } |Cd_{b,p} - Cd_{b',p}| \leq 7dBm \}$$

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Ensuite, la contrainte à respecter pour chaque cellule est la suivante :

$$\forall b \in BS \text{ handCell}(b) \neq \emptyset.$$

b. Trafic

En plus de la contrainte de couverture, un réseau GSM doit garantir aussi un certain trafic. Le trafic à assurer est on fonction du trafic de chaque zone. Le trafic peut être estimé par des méthodes statistiques et dépend du type de la zone (zone rurale, zone industrielle...) et de la densité de la population. Par exemple le trafic dans une zone rurale est inférieur à celui d'une zone industrielle.

Dans cette modélisation où le trafic attendu dans chaque point de TT est donné, il peut être calculé pour chaque station de base par la formule suivante:

$$\text{Traffic_BS}(b) = \sum_{p \in TT \cap \text{Cell}(b)} \text{traffic_point}(p)$$

Le nombre de TRXs qui doivent être présent sur une station de base est ensuite déduit en fonction de Traffic_BS ,

Le nombre de trafic pris en charge par une BS ne peut dépasser 43 Erlang (BS contient 7 TRXs), c'est pour cela qu'on a la formule suivante qui désigne le trafic pris en charge:

$$\text{TrafficHold}(b) = \begin{cases} \text{Traffic_BS}(b), & \text{si le } \text{Traffic_BS}(b) \leq 43 \\ 43, & \text{sinon} \end{cases}$$

Ainsi, la fonction à maximiser est la suivante:

$$f = \sum_{b \in BS} \text{trafficHold}(b)$$

c. Le rapport Capacité prise en charge/ Capacité station de base

Permet de donner une idée sur le trafic pris en charge par rapport au trafic maximal qui peut être supporté par une BS (noté trafficYield).

$$\text{trafficYield}(b) = \frac{\text{trafficHold}(b)}{\text{trafficCapacity}(b)}$$

Il faudra donc maximiser la fonction suivante :

$$f = \sum_{b \in BS} \text{trafficYield}(b)$$

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

d. Nombre de sites utilisés

Un des principaux objectifs est de minimiser le nombre de stations de sites utilisés.

$$\text{Min } \sum_{i \in L} c_i \times y_i, \quad y_i = \begin{cases} 1, & \text{si le site } i \text{ est utilisé} \\ 0, & \text{sinon} \end{cases}$$

c_i est le cout de déploiement du site i , il peut être pris égal à 1 pour tous les sites.

e. Couverture

Tous les points (ou bien un nombre de points prédéfinis) doivent être couverts. En termes signal radio, ces points doivent être couverts par une puissance du signal radio supérieure au seuil minimum.

$$\text{Cover} = \bigcup_{b \in \text{BSI}} \text{Cell}(b)$$

Si la couverture est considérée comme objectif alors la fonction Cover doit être maximisée. Par contre, si elle est prise comme contrainte, c.à.d. tous les points STP doivent être couverts, alors Cover doit être égale à l'ensemble ST.

f. Bruit et interférences

Souvent un point STP dans la zone couverte reçoit plus d'un signal de la part de plusieurs BS, par contre une cellule est un ensemble de ces points mais qui reçoivent tous comme meilleur signal $Cd_{b,p}$ de la part de la même BS b . On ne peut pas avoir tous les points STP recevant qu'un seul signal puisque les autres signaux sont nécessaires pour assurer le *handover*. Toute fois, ça serai bien de limiter le nombre de signaux reçus ayant une puissance supérieure à un seuil S_m donné (le seuil pour qu'un signal soit senti par la station mobile) et inférieure à $Cd_{b,p}$.

Soit : $Cd_{b,p} \geq Cd_{b_1,p} \geq \dots \geq Cd_{b_h,p} \geq \dots \geq Cd_{b_k,p} > S_m$, la puissance des signaux reçus par un STP p , le niveau de bruit est donné par la formule :

$$\text{Noise}(p) = \sum_{h < j \leq k} Cd_{bj,p} - S_m$$

h est le nombre autorisé de signaux couvrant le point p [37].

Donc, la fonction à minimiser est donnée par:

$$f = \sum_{p \in ST} noise(p)$$

Cette modélisation permet de refléter une vue plus au moins réelle et détaillée du problème, ce qui permet d'extraire plusieurs fonctions qui reflètent les différents objectifs et contraintes.

3.2.2 Modélisation de Calegari et al

Calegari et al ont aussi proposé un modèle dans [10] pour décrire la zone de travail. Celle-ci est discrétisée, C.à.d., décrite par une collection de locations géographique finie. Cette collection est définie à l'aide d'une grille carrée référencé géographiquement. Les points de la zone sont les points d'intersection des lignes de la grille.

L'ensemble des points de la zone est considéré comme deux sous-ensembles, l'ensemble L de tous les emplacements à couvrir, et l'ensemble M de tous les emplacements potentiels pour installer des transmetteurs (BS). Soit G le graphe $(M \cup L, E)$, où E est un ensemble d'arcs tel que chaque transmetteur est lié aux emplacements qu'il couvre.

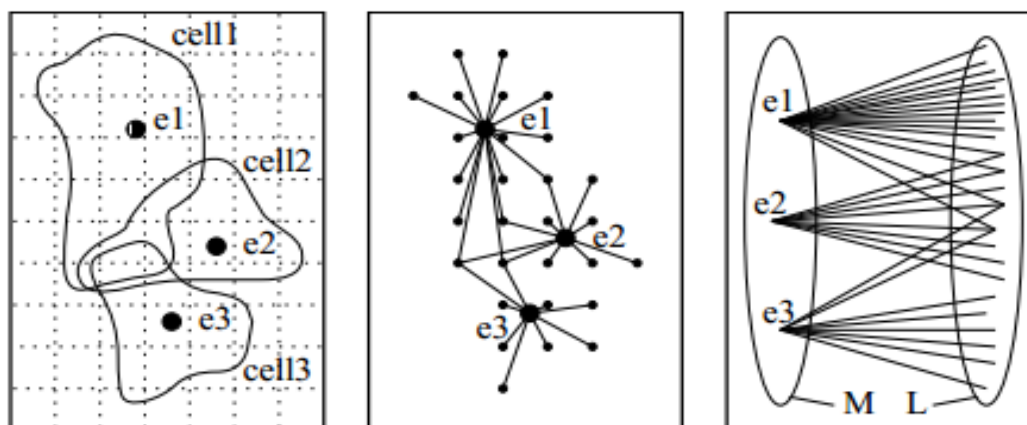


Figure 1.5 : (a) Trois transmetteurs potentiels e_1 , e_2 , e_3 et leurs cellules discrétisées. (b) Un graphe où les arcs lient les transmetteurs aux emplacements qu'ils couvrent, est conçu. (c) Le graphe est représenté comme un graphe biparti, où M est l'ensemble des emplacements potentiels des transmetteurs et L est l'ensemble des emplacements à couvrir.

Le problème revient à chercher un sous-ensemble minimum de transmetteurs qui couvrent une surface maximum de la zone à couvrir. Formellement, on cherche un sous ensemble $M' \subseteq M$ tel que $|M'|$ est minimum, et $|\text{Neighbours}(M',E)|$ est maximum, tel que :

$$\text{Neighbours}(M',E) = \{u \in L \mid \exists v \in M', (v,u) \in E\}.$$

Divers travaux ont opté pour cette modélisation, à savoir [6] où il a utilisé l'algorithme MOCHC pour la phase de résolution. [25] a amélioré cette méthode puisque le graphe étant bi-partie ne permet pas de représenter la couverture des points qui appartiennent à l'ensemble L, c.à.d. les points qui ont les mêmes positions que les stations de base. Donc il a proposé une autre modélisation basée sur les hypergraphes.

3.2.3 Modélisation de Mendes

Mendes et al [25] propose une modélisation basée sur celle de [10] mais modifiée pour palier au problème cité ci-dessus. La zone est modélisée par un hypergraphe $H=(V,E)$, où $V= \{v_1,v_2,\dots, v_n\}$ est un ensemble non vide et $E=E_1,E_2,\dots,E_m$ est un ensemble de combinaisons non vides de V, tel que $V= \cup E_i$.

Donc H est un ensemble de graphes dirigés $p(x) = x^E$, où les arcs sont dirigés de x (le centre) vers un ensemble d'éléments de E. Où x est le sommet centrale qui représente une station de base.

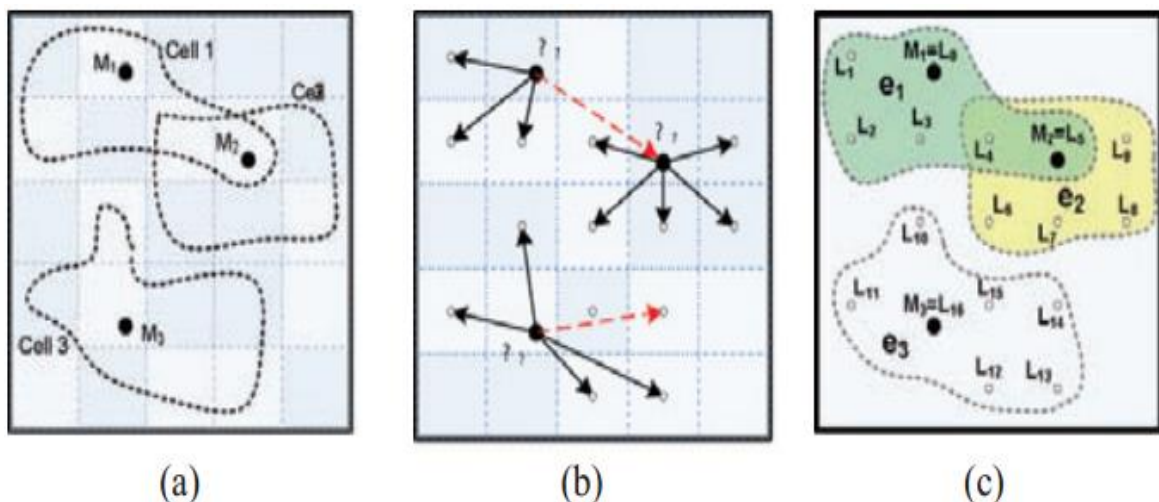


Figure 1.6 : Exemple de Trois stations de bases, (a) leurs cellules associées (b) les graphes qui relient chaque transmetteur par les points qu'il couvre, un arc coloré et utilisé pour relier les transmetteurs couvert, et (c) la représentation de l'hypergraphe associé.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Cette modélisation permet de prendre en compte les points qui appartiennent à M et L en même temps c.à.d. les points qui se trouvent au même endroit qu'un transmetteur.

Afin d'évaluer la qualité d'un ensemble H' de BS, une fonction objectif f largement utilisée dans le domaine de la télécommunication a été choisie:

$$f(x) = \frac{CoverRate(x)^2}{Number\ Transmitters\ Used(x)}$$

où :

$$CoverRate(x) = 100 \frac{|Neighbors(M', E')|}{|Neighbors(M, E)|}$$

Et:

$$Neighbors(M', E') = \{u \in L | \exists v \in M', (u, v) \in E\}$$

Dans cette modélisation seulement deux objectifs sont pris en compte, maximiser le degré de couverture réseau et minimiser le coût de l'installation des antennes.

Reininger	Calégari et Mendes
Plus proche du monde réel	Fait abstraction à plusieurs problèmes techniques
Traite un nombre important d'objectifs et contraintes	Deux objectifs seulement (couverture, nombre d'antennes utilisés)
Complexe et nécessite une grande quantité de calculs	Simple
Résolution de vraies instances	Simplification du problème

Tableau 1.1 : Comparaison *Reininger et al*, *Calégari et al* et *Mendes et al*

La première modélisation de *Reininger et al* [29] présentée, offre une meilleure vue sur les caractéristiques techniques du problème par rapport aux deux autres

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

modélisations. Par contre, les autres modélisations offrent une vue simplifiée du problème ce qui rend le problème à résoudre plus simple.

3.3 Modèles de propagations d'ondes

Les modèles de propagation d'ondes sont des modèles mathématiques qui permettent de prédire l'impact d'un émetteur sur la zone de réception soit la puissance du signal reçu. Ces modèles prennent en charge certains paramètres qui caractérisent l'environnement en plus de la puissance signal émise par le transmetteur pour prédire la puissance du signal reçu. On peut citer comme exemple le modèle *Free Space* qui est largement utilisé, qui suppose qu'il n'y a pas d'obstacles entre l'émetteur et le récepteur et que l'environnement est sans bruit. Ce modèle a été utilisé dans [16].

3.4 Complexité du problème

Le problème APP rappelle le problème USCP (*unicost set covering problem*) connu pour être NP-Difficile, où étant donnée une famille de sous-ensembles d'un ensemble A , on cherche à reconstituer l'ensemble A par l'union d'un nombre minimum de ces sous-ensembles.

Par contre le problème APP diffère par rapport au problème USCP dans le but à atteindre, où dans l'APP on ne cherche pas à trouver la solution qui assure la couverture total mais on peut se contenter d'une solution qui offre une bonne couverture. Le problème APP est un problème d'optimisation combinatoire NP-Difficile [10].

Le problème APP a été abordé par plusieurs approches d'optimisation, soit par des méthodes exactes comme la méthode (ex : branch and bound [7] et la programmation linéaire) soit par des méthodes approchées (les méta-heuristiques). Dans ce type de problèmes, caractérisé par un grand espace de recherche, un compromis entre le temps de réponse et l'exactitude de la solution doit avoir lieu.

La plupart des problèmes d'optimisation liés aux réseaux GSM (FAP, APP...) sont par contre inabordable par les méthodes exactes, d'où l'utilisation des approches heuristiques [5]. En fait, on préfère généralement gagner en efficacité et avoir plutôt des solutions approchées. Ce qui revient à dire qu'il convient d'opter pour le deuxième type d'approches.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

4 Conclusion

Nous avons introduit dans ce chapitre les différentes notions concernant les réseaux GSM, la modélisation du problème et sa complexité. Le chapitre suivant sera consacré aux méthodes de résolutions de ce type de problème, à savoir, les métaheuristiques.

Chapitre 2 :

Méthodes de résolution approchées (Approches évolutionnaires et approches de voisinage)

1 Introduction

L'APP a été abordé dans plusieurs travaux où, généralement après la modélisation, le processus d'optimisation (recherche de solutions qui optimisent les objectifs) est le cœur de la résolution du problème. Comme illustré dans le chapitre précédent les méthodes d'optimisation approchées sont les mieux adaptées pour ce cas.

Dans ce chapitre nous allons faire le tour sur les différentes méthodes de résolution et travaux qui ont été réalisés dans ce domaine.

Malgré que le problème APP est un problème d'optimisation multiobjectif, beaucoup de méthodes le transforment en un problème mono objectif avant de le traiter. Sauf qu'il existe aussi des méthodes de résolution qui traitent le problème directement c'est-à-dire des méthodes multiobjectif.

2 Travaux antérieurs

Plusieurs travaux ont été réalisés pour la résolution du problème de positionnement d'antennes, utilisons différentes modélisations et approches. On peut citer comme exemple le travail de *Reininger et al* [29] où il a proposé une nouvelle modélisation (décrite dans le chapitre précédent) qui prend en considération plusieurs aspects réelle du problème représentés sous forme d'objectifs et contraintes. Cette modélisation est caractérisée par l'importance de la complexité de résolution de ses instances. En se basant sur cette modélisation *Vasquez et al* dans [37] ont proposé une approche en trois phases utilisant la recherche taboue. Où la phase de prétraitement vise à éliminer de l'espace de recherche les solutions dont il est possible de savoir auparavant qu'elles ne peuvent pas être de bonne qualité, comme les solutions qui emploient un nombre très petit d'antennes. La phase recherche est basée sur l'algorithme de recherche taboue. Enfin une autre phase de post traitement qui consiste à régler le paramétrage des antennes. *El-Ghazali et al* [16] s'est aussi fondé sur cette modélisation dans son travail, où il a présenté des modèles hiérarchiques parallèle et il a montré que l'utilisation de ces derniers peut accélérer la recherche pour la résolution du problème et donner de meilleures solutions. Il a employé un algorithme génétique avec une résolution multiobjectif.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Dans [10], une autre modélisation a été définie par *Calegari et al.* Cette modélisation est plus simple que celle de *Reininger et al* [29], car elle fait abstraction à plusieurs aspects techniques du problème comme les paramètres d'antennes et les stations mobiles, ce qui rend le problème à résoudre nettement moins complexe.

Le problème APP a été considéré comme un problème mono objectif dans plusieurs travaux, où une seule fonction objectif est à optimiser. Des algorithmes mono objectif ont été utilisés pour la résolution, comme dans le travail de *Alba et al* [3] où ils ont utilisé l'algorithme CHC, ou dans *Alba et al* [4] où un algorithme génétique parallèle a été utilisé. Par contre, le problème APP a aussi été considéré comme problème multiobjectif, où des algorithmes de résolution multiobjectif ont été employés comme l'algorithme MOCHC et l'algorithme NSGA-II dans le travail d'*Antonio et al* [6], et l'algorithme génétique multiobjectif dans le travail de *Meunier* [26].

Une grande panoplie d'algorithmes ont aussi été employés dans *Mendes et al*[25], et ont aussi été comparés sur la base de mesures normalisées afin d'abstraire tout aspect technique indépendant du problème lui-même, comme l'environnement d'exécution et le matériel utilisé.

3 APP comme problème d'optimisation

Comme le problème APP est un problème d'optimisation, pour définir ses principaux composants on définit les éléments qui caractérisent tous problèmes d'optimisation:

Une approche mono objectif permet de rendre le problème moins complexe, mais son inconvénient est qu'elle ne rend que la solution qui a la meilleure valeur de la fonction objectif et ne permet pas de donner un choix à faire pour l'utilisateur entre plusieurs solutions afin qu'il puisse faire un compromis lui-même sur les objectifs à optimiser. En plus, concevoir une fonction d'évaluation qui représentera tous les objectifs n'est pas toujours évident.

3.1 Espace de recherche (de décision) :

Représente l'ensemble de solutions ou de configurations possible, constitué des différentes valeurs prises par les variables de décision.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

3.2 Fonction objectif :

Une ou plusieurs fonctions à optimiser (minimiser ou maximiser) qui reflètent la qualité de la solution. Lors que le problème est considéré multiobjectif, des solutions qui optimisent tous les objectifs à la fois existent rarement, on essaye par fois de faire des compromis entre les objectifs. Mais lorsque le problème est considéré comme mono objectif, les différents objectifs sont agrégés dans une seule fonction objectif.

Les objectifs et les contraintes qui concernent notre problème sont définis dans le chapitre précédent. Toute fois la plupart des travaux consacrés à ce domaine ne traitent pas tous les objectifs et les contraintes.

Le problème consiste à chercher dans l'espace des solutions réalisables la solution (ou l'ensemble de solutions) qui rend (rendent) la (les) meilleure (s) valeur (s) de la fonction objectif. Le problème peut se ramener à chercher à optimiser plusieurs objectifs, ce qui le rend de nature multiobjectif. Mais il peut aussi être considéré comme problème mono objectif, comme dans *Calegari et al* [10], où une seule fonction objectif a été définie. Cette fonction englobe les différents objectifs de l'optimisation. La fonction f a été reprise par la suite dans beaucoup de travaux concernant l'APP.

$$f = \frac{\text{taux de couverture}^\alpha}{\text{nombre d'antennes}}$$

α permet de définir une priorité entre la qualité de la couverture par rapport au cout (nombre d'antennes). Elle est généralement prise égale à 2. Ce qui avantage les solutions qui offrent une bonne couverture aux solutions qui utilisent un nombre réduit d'antennes mais qui n'assurent pas une bonne couverture.

3.3 Contraintes:

Généralement dans un problème d'optimisation il y'a toujours un ensemble de contraintes à respecter par les solutions. Ces contraintes peuvent être des contraintes d'inégalité ou d'égalité. Le *handover* expliqué précédemment on est une par exemple.

3.4 Codage de solution

Les métaheuristiques travaillent avec des solutions codées selon différents types de codage. De même pour les variables de ces solutions qui peuvent être de nature

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

diverses (réelle, entier, booléenne, etc.) et exprimer des données qualitatives ou quantitatives.

Généralement, les variables de l'APP sont codées dans des vecteurs binaires. Une solution est généralement codée dans un vecteur où chaque élément du vecteur est associé avec une station de base. Un élément du vecteur prenant 1 comme valeur signifie que la station de base associée est utilisée. La valeur 0 signifie le contraire. Cette représentation est utilisée dans [6] (Voir figure 2.3-a). Dans certains travaux où l'on inclut le paramétrage des antennes dans le processus d'optimisation, donc dans le codage de la solution, une représentation hiérarchique est proposée. Dans cette représentation une solution est constituée de plusieurs niveaux. Dans la figure 2.3 (b), un exemple où le premier niveau indique si le site est activé ou non. Le deuxième niveau indique le type d'antenne utilisé. Si l'antenne est de type omnidirectionnel nous n'avons besoin que d'un niveau supplémentaire pour stocker les paramètres *Power* et *Tilt*. Dans le cas où l'antenne est de type Directionnel, trois niveaux supplémentaires sont nécessaires, le numéro de la station, le type, et les paramètres (Trois stations au maximum). Cette dernière représentation a été utilisée dans [16] et [37]

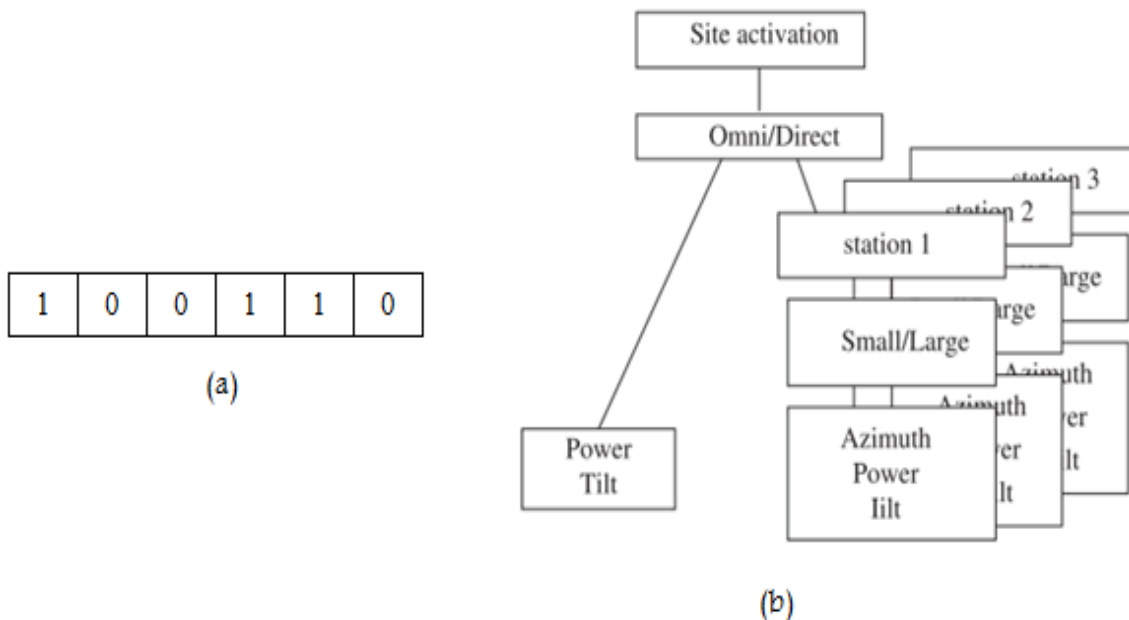


Figure 2.1 : (a) Exemple d'une représentation binaire, (b) exemple d'une représentation hiérarchique.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Les travaux réalisés pour résoudre le problème de l'APP ont généralement proposés de le résoudre par des méthodes soit évolutionnaires soit par des méthodes de voisinages [26] [37]. Ces méthodes seront l'objet de ce chapitre. Le reste de ce chapitre traitera premièrement les approches multiobjectif et les travaux réalisés concernant celles-ci. Ensuite, il traitera aussi les approches mono objectif et les travaux concernant celles-ci aussi.

4 Optimisation multiobjectif

Nous donnons ci-dessous quelques notions dont sont basées les méthodes d'optimisation multiobjectif lors que l'APP est considéré comme un problème multiobjectif. En effet, le problème étant d'optimiser plusieurs objectifs (couverture, nombre de station de base...) à la fois, il existe des approches qui tentent de traiter ces objectifs simultanément.

Formellement, un problème d'optimisation multiobjectif est décrit comme suit :

$$\begin{cases} \text{Optimiser } f_m(x), & m = 1, \dots, M; \\ g_j \geq 0, & j = 1, \dots, K; \\ h_j = 0, & j = K + 1, \dots, L; \\ x_i^l \leq x_i \leq x_i^u, & i = 1, \dots, n. \end{cases}$$

L'optimisation du vecteur objectif $f(x) = (f_1(x), f_2(x) \dots f_m(x))$ revient soit à minimiser ou à maximiser l'ensemble de toutes les fonctions objectifs ou minimiser certaines fonctions objectifs toute en maximisant d'autres comme dans notre cas où des objectifs sont à maximiser (couverture) et d'autres sont à minimiser (coût). Dans ce dernier cas, pour rendre tous les objectifs à minimiser (à maximiser), les fonctions objectifs à maximiser (à minimiser) sont transformées en fonctions à minimiser (à maximiser) par multiplication par -1 .

$$\text{Maximiser } f(x) \equiv \text{Minimiser } -f(x)$$

x est un vecteur de n variables de décisions (de type binaire dans notre cas), $x=(x_1, x_2, \dots, x_n)$. L'espace de décision $S = \{0,1\}^n$.

Souvent, les objectifs présentent un conflit entre eux, ce qui rend très improbable de trouver une solution qui optimise tous les objectifs à la fois. En effet,

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

les résultats d'une telle approche ne constituent pas un ordre total. Ni au moins, le but est de fournir une variété de solutions qui vont permettre de faire des compromis entre les objectifs.

Lorsqu'il y a plusieurs objectifs, la notion d'optimum change, elle est remplacée par les notions de dominance et d'optimalité de Pareto.

On introduit dans ce qui suit quelques définitions pour illustrer la notion d'optimalité dans le cas multiobjectif, basés sur les travaux de Pareto :

Définition 1: Soit un ensemble S de solutions, et m fonctions objectifs $F = \{f_1, f_2, \dots, f_m\}$, Soient s_1 et s_2 deux solutions de S : On dit que s_1 domine s_2 (on note $s_1 < s_2$) ssi:

$$\forall f \in F, f(s_1) \leq f(s_2)$$

$$\exists f \in F, f(s_1) < f(s_2).$$

Définition 2: On définit ainsi l'ensemble Pareto S^* par :

$$S^* = \{s \in S \mid \nexists s' \in S : s' < s\}$$

Définition 3: Le front optimal de Pareto est défini par :

$$F_{S^*} = \{f(s) \in \mathbb{R}^m \mid s \in S^*\}$$

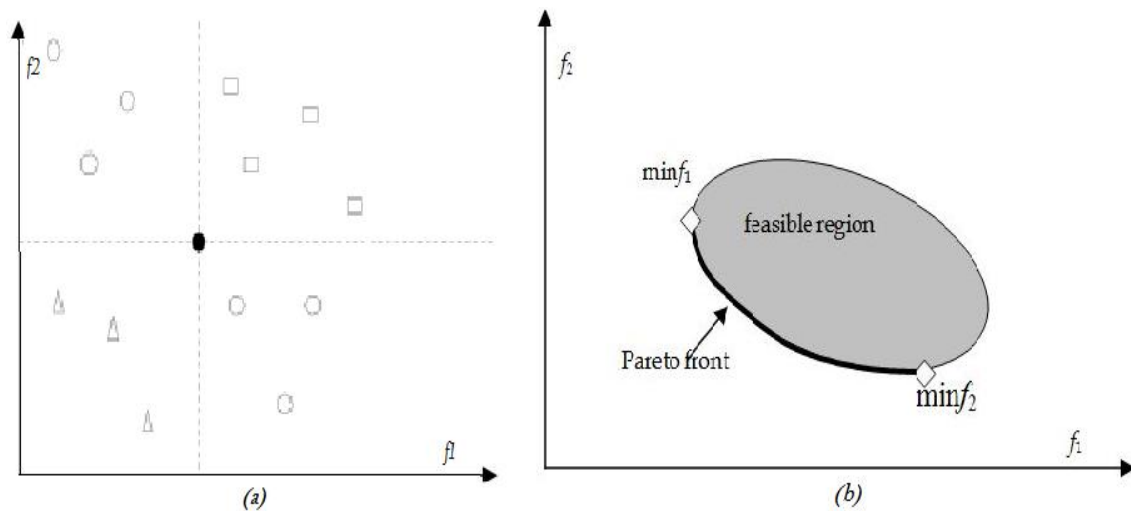


Figure 2.2 : (a) Le point noir est dominé par chacun des triangles, domine chacun des rectangles et est équivalent aux anneaux au sens de la dominance. (b) Le front optimal de Pareto représenté par la ligne en gras.

Deux objectifs principaux sont attendus des résultats fournis par une optimisation multiobjectif :

Convergent vers le front Pareto. En effet, la plupart des recherches sur l'application des algorithmes génétiques à des problèmes multi objectifs se concentrent sur l'étape de sélection. Dans cette étape, des méthodes de classement sont appliquées pour attribuer des performances aux individus. La performance dépend de la notion de dominance, et donc dépend directement de l'optimalité Pareto.

Les solutions doivent être diversifiées toute au long du front Pareto et ne doivent pas se concentrer dans une niche principale (optimum globale).

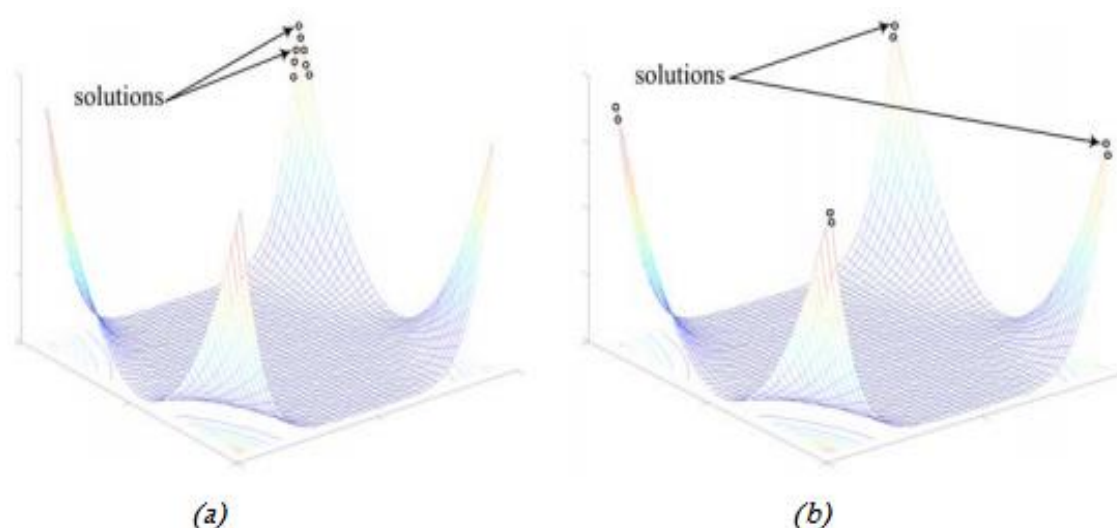


Figure 2.3: (a) Solutions concentrées dans l'optimum global, (b) solutions diversifiées.

5 Méthodes d'optimisations (Méthodes de résolutions)

Les espaces de décisions des problèmes réels sont caractérisés par leurs grandes tailles. Ainsi, généralement il est préférable d'avoir recourt à une résolution approchée, qui consiste à trouver les solutions les plus proches possible de l'optimum au lieu d'une résolution exacte. Comme illustré dans la section 3.4, l'APP est un problème NP-Difficile, où les méthodes les mieux adaptées et les plus appliquées pour ce type de problèmes sont les approches heuristiques [5]. Ce qui nous ramène à notre objet d'étude dans ce chapitre qui porte sur les métaheuristiques.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

5.1 Définition des métaheuristiques

Formellement, Les métas heuristiques sont définis comme des processus de génération itératifs qui guident une heuristique subordonnée en combinant intelligemment différents concepts pour l'exploration et l'exploitation de l'espace de recherche afin de trouver des solutions optimales[27].

5.2 Classification des métaheuristiques

Selon le critère pris en compte, la classification peut être différente. Si l'on considère par exemple comme critère le fait que l'algorithme s'inspire ou non de phénomènes naturels. Par exemple, les algorithmes génétiques et les algorithmes de fourmis s'inspirent respectivement de la théorie de l'évolution et du comportement de fourmis à la recherche de nourriture. Par contre, ce n'est pas le cas pour la méthode Tabou.

D'autres classifications se basent sur le nombre de structures de voisinages (La fonction N) utilisées. Étant donné qu'un optimum local relativement à un type de voisinage ne l'est pas forcément pour un autre type de voisinage, il peut être intéressant d'utiliser des métaheuristiques basées sur plusieurs types de voisinages. Un autre critère peut être considéré, et qui consiste en l'utilisation ou non de mémoire (historique de la recherche).

Une autre façon de classer les métaheuristiques est de distinguer celles qui travaillent avec une population de solutions de celles qui ne manipulent qu'une seule solution à la fois. Les méthodes qui tentent itérativement d'améliorer une solution sont appelées méthodes de recherche locale ou méthodes de trajectoire. La méthode Tabou, le Recuit Simulé et la Recherche à Voisinages Variables sont des exemples typiques de méthodes de trajectoire. Ces méthodes construisent une trajectoire dans l'espace des solutions en tentant de se diriger vers des solutions optimales. L'exemple le plus connu de méthode qui travaille avec une population de solutions sont les algorithmes génétiques.

Toutes ces classes peuvent s'adapter aux problèmes d'optimisation mono objectif ainsi qu'aux problèmes multiobjectif. Et comme le problème APP peut être considéré selon les deux approches, nous présenterons dans la suite, les deux approches mono et multiobjectif ainsi que les classes des algorithmes évolutionnaires et des algorithmes de voisinage.

6 Approche mono objectif

Comme nous l'avons déjà défini dans la section 3, il s'agit dans cette approche d'optimiser qu'une seule fonction objectif. Nous présenterons donc quelques algorithmes des deux classes évolutionnaires et de trajectoires.

6.1 Métaheuristiques évolutionnaires (EA)

Les méthodes évolutionnaire sont des méthodes qui manipulent un ensemble de solutions simultanément. Elles font évoluer une population d'individus selon des règles bien précises. Ces méthodes améliorent au fur et à mesure des itérations une population de solution.

Ces méthodes exécutent des procédures d'adaptation individuelle et des procédures de coopération (échange d'information entre individus) jusqu'à ce qu'un critère d'arrêt soit satisfait (par exemple un nombre d'itération franchit sans qu'il y'est une amélioration).

Différentes caractéristiques permettent de faire la différence entre diverses méthodes évolutives. Par exemple le type d'individu qui n'est pas forcément une solution. En effet, il peut être une partie de la solution, ou bien d'objets que l'on peut transformer en solution. Le type d'évolution ou bien la manière de remplacement de la population, ie, *remplacement générationnel* où toute la population est remplacée ou bien remplacement stationnaire (steady state) où seulement une partie de la population qui est remplacée.

Algorithme : Structure générale d'un algorithme évolutionnaire

Début

$t = 0$

Initialiser et Evaluer (P(t))

Tant que (condition d'arrêt non satisfaite) **faire**

$P'(t) = \text{variation } [P(t)]$

Evaluer $[P'(t)]$

$P(t+1) = \text{select } [P'(t), P(t)]$

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

t =t+1

Fait

Fin

Des méthodes appartenant à cette classe de métaheuristiques (EA) ont été implémentées afin de résoudre le problème APP. En fait, la grande partie des travaux qui ont été réalisés portent sur les méthodes évolutionnaires [10], [2] Cette section est consacrée à l'étude de ces méthodes et des résultats des travaux qui ont implémentés ces méthodes.

6.1.1 Les algorithmes génétiques

Les algorithmes génétiques sont basés sur le concept de la sélection naturelle élaborée par Charles Darwin. En effet, ces algorithmes adoptent une sorte d'évolution artificielle analogue à l'évolution naturelle. Il s'agit de simuler l'évolution d'une population d'individus diverse à laquelle on applique différents opérateurs d'évolution comme définit en dessous:

- **croisement** (mélange du matériel génétique) : combinaison de deux individus pour engendrer un ou deux nouveaux individus
- **mutation** (perturbation du matériel génétique) : modification aléatoire d'un individu
- **sélection** (basé sur l'évaluation des individus - **fonction d'évaluation**) : choix des individus formant la nouvelle population.

Nous parlerons donc d'*individus*, pour parler de solutions. L'ensemble des individus formera une *population*, qui est évoluée pendant un certain nombre d'itérations appelées *générations*, jusqu'à ce qu'un critère d'arrêt soit vérifié. Pour passer d'une génération à une autre, la population est soumise à des *opérateurs de sélection*. Les *opérateurs de variation* (Croisement, mutation), eux, permettront de transformer la population, de façon à favoriser l'émergence de meilleurs individus. Une solution est codée dans une structure appelée *chromosome*. Ce dernier est constitué d'une série de gènes (élément d'un génotype). Une variante d'un gène est appelée *Allèle*. Dans notre cas, chaque gène donne des informations concernant un site donné (utilisé ou non, configurations des paramètres...).

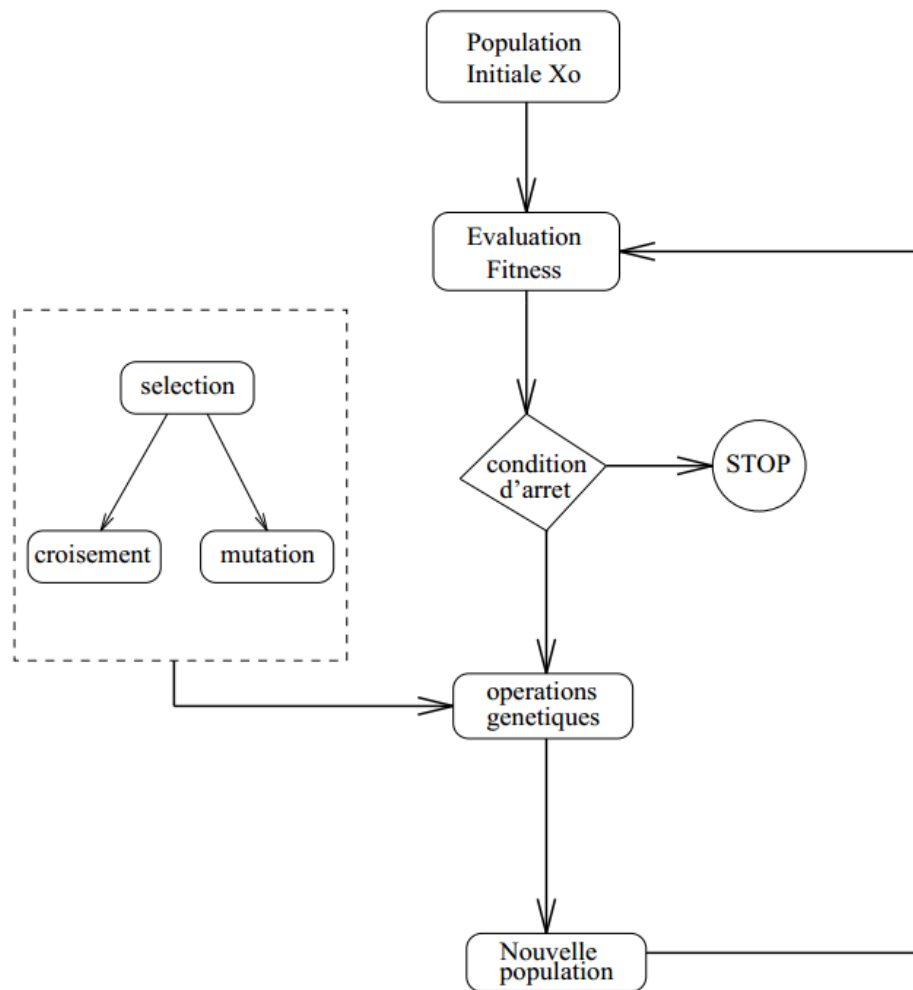


Figure 2.4 : Organigramme d'un algorithme génétique

6.1.2 Principaux étapes d'un algorithme génétique

L'algorithme génétique inclut les étapes suivantes :

- Génération aléatoire d'une population d'individus
- Evaluation de l'ensemble des individus de la population initiale.
- Sélection d'un certain nombre d'individus dans la population, afin de produire une population intermédiaire, appelée aussi « mating pool ».
- Sélection de deux chromosomes parents P1 et P2 en fonction de leurs adaptations. On applique aléatoirement l'opérateur de croisement avec une probabilité P_c pour générer deux chromosomes enfants C1 et C2. L'opération de croisement est suivie par l'opération de mutation avec une probabilité P_m , ce qui produit deux nouveaux individus C'1, C'2 pour lesquelles leurs fonctions fitness sont évaluées avant qu'ils soient insérés dans la nouvelle population. L'opération

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

de croisement favorise l'intensification de la recherche, par contre la mutation permet l'obtention de résultats diversifiés.

- Les opérations de croisement, mutation et de sélection sont réitérées jusqu'à satisfaction du critère d'arrêt.

6.1.3 Algorithmes génétique générationnel et stationnaire

Ces deux types diffèrent dans l'étape de reproduction, plus précisément du nombre d'individus générés pour remplacer les anciens individus. Un algorithme génétique générationnel noté **gGA** remplace toute une population par une nouvelle de taille λ . Par contre, les algorithmes de type stationnaire (*steady state*) noté **ssGA**, un ou deux nouveaux enfants seulement sont créés et insérés. Ces deux types d'algorithmes sont des sous classe d'une classe plus large d'algorithmes qui sont caractérisés par leur pourcentage d'individus remplacés dans la population.

6.1.4 Le problème APP résolu avec ssGA et gGa

En fait, les algorithmes ssGA et gGa sont considérés comme des algorithmes évolutionnaires classiques où généralement on essaye de comparer les résultats des nouveaux algorithmes appliqués à un problème avec ces derniers. Comme c'est le cas aussi pour le problème de l'APP, où par exemple on cite le travail de [3] qui a implémenté l'algorithme CHC et a comparé les résultats de ce dernier avec les résultats obtenus par ssGA et gGA. Cette méthode sera illustrée dans la section suivante.

On peut citer aussi un travail qui est plus ancien de [10] qui a essayé de remédier au problème de lenteur qui caractérise ces algorithmes génétiques qui est due dans son cas à la taille énorme de son graphe bipartite de ça modélisation (décrite dans le chapitre 1) et due aussi à la charge de calcul importante causé par le besoin d'avoir une grand taille de population afin que les résultats soient d'une bonne qualité. Ainsi dans ce travail l'auteur à étudié ces algorithmes afin de proposer une technique qui permet l'implémentation parallèles de ces algorithmes. Nous étudierons aussi les travaux et les techniques ou modèle des algorithmes parallèles plus loin.

6.1.5 Algorithme CHC

L'algorithme CHC (Cross-generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation) a été proposé par Eshelman [14]. Cet

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

l'algorithme comporte une combinaison entre une stratégie de sélection avec une grande pression de sélection et d'autres composants pour offrir une grande diversité des solutions. En effet, CHC a permis d'obtenir de très bons résultats [3] [28] mais, contre toute attente, il n'a pas beaucoup été utilisé.

L'algorithme travaille avec une population d'individus qu'on note P_a et à chaque étape un nouvel ensemble de solutions P_n est produit par sélection de paires à partir de la population et les recombine.

Afin de garder la population diversifiée et d'empêcher la convergence prématurée, l'algorithme CHC combine les éléments suivants :

- Une sélection élitiste : Les M membres de l'ancienne génération (parents) sont mélangés avec les individus résultants (enfants) de cette dernière pour ensuite choisir les M meilleurs individus pour former la nouvelle population.
- Un opérateur de croisement perturbateur (HUX, Half Uniform Crossover) qui fait échanger entre les parents la moitié des allèles (gènes) qui sont différents. Ceci garantit que les deux offspring (enfants) sont toujours à la même distance de Hamming (nombre de bits différents) de leurs parents, ainsi une nouvelle population hautement diversifiée est obtenue.

On remarque bien l'équilibre offert par ces deux opérations, où la première permet la convergence, par contre la deuxième permet la prévention contre la convergence prématurée. En plus, CHC utilise aussi les mécanismes suivants:

- Mécanisme de prévention contre l'inceste (*incest prevention mechanism*): Malgré que la sélection des parents certes ce fait aléatoirement, mais la recombinaison ne se fait que dans le cas où les parents ne sont pas très semblables (la distance de Hamming entre eux est supérieure à un certain seuil).
- Une procédure de redémarrage (*restart process*): cette procédure remplace l'opération de mutation et est appliquée lorsque la population a convergé. Après un certain nombre de générations sans amélioration, la population est réinitialisée en gardant le meilleur individu dans la nouvelle population et on ajoute les $M-1$ individus qui manquent par changement aléatoire de quelques bits de la meilleure solution.

Algorithme CHC

Début

t = 0

Initialiser (Pa, nb_convergence)

Tant que (condition d'arrêt non satisfaite) **faire**

Parents= Sélection-parents(Pa)

Offspring=HUX(Parents)

Evaluer (Offspring)

Pn= Sélection-Elitiste(Offspring, Pa)

Si (identique(Pa, Pn)) **alors**

Nb_convergence = Nb_convergence - 1

Si (Nb_convergence == 0) **alors**

Pn= redémarrer(Pa)

Initialiser(Nb_convergence)

FinSi

FinSi

t = t + 1

Pa = Pn

Fait

Fin

La variable “convergence-count“ représente le nombre de générations sans amélioration avant d'effectuer un redémarrage du processus.

6.1.6 APP résolu avec CHC

L'algorithme CHC ayant rendu de bons résultats mais pas beaucoup utilisé, Alba et Al [3] ont voulu voir ce que ça donnait en étant appliqué au problème APP. L'algorithme a été comparé contre les algorithmes gGA et ssGA on plus de l'algorithme dssGA qui est une version distribuée de l'algorithme ssGA. Ces algorithmes ont été appliqués en utilisant la modélisation de calégari (décrite dans le chapitre précédent) et 3 différents types d'antennes qui sont caractérisées par la

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

forme de la cellule formée par ces derniers (des antennes qui forment des cellules carrés, des antennes omnidirectionnelles qui forment des cellules circulaires et des antennes directionnels). Plusieurs expérimentations ont été réalisées. Dans un premier lieu, des antennes de types carré et circulaire ont été utilisées chacun dans une expérimentation, et tel que chaque instance est résolue par tous les algorithmes. CHC semble être celui qui a fourni les meilleurs résultats comparé aux algorithmes cités en plus de l'algorithme SA qui est un algorithme basé sur la recherche locale (sera traité dans la partie qui traite sur les méthodes de recherche locale). Ensuite, dans la deuxième partie des expérimentations, CHC a été testé seul sur d'autres variantes du problème où cette fois ci, une fois des antennes directionnels ont été utilisés est paramétrés (azimuth qui détermine la surface couverte). Et une autre fois, tous les types d'antennes ont été utilisées dans la même instance. Cette deuxième partie d'expérimentation avait comme but d'étudier le comportement de l'algorithme CHC lorsque la taille du problème accroît. Le cout de résolution de ces instances s'est avéré accroître de manière sous-exponentiel (croît plus lentement que toute exponentielle).

6.1.7 Algorithme PBIL

PBIL (Population-based incremental learning) est une méthode qui combine les algorithmes génétiques avec un apprentissage non supervisé à savoir l'apprentissage compétitif (réseau de neurones).

L'algorithme PBIL agit sur une population d'individus et construit une nouvelle génération à chaque itération. Chaque solution d'une génération est construite à l'aide d'un vecteur de probabilité P . Les valeurs d'un vecteur solution V étant soit égale à 1 soit à 0 (station de base activée ou non dans notre cas) une probabilité de P_i élevée indique qu'il y'a plus de chance pour V_i (station de base i) d'être activée. Dans le cas contraire (la valeur de P_i est petite) la valeur de V_i a plus de chance d'avoir 0 comme valeur (station de base i mise hors tension).

Au début, une première génération est générée par des valeurs de P égale à 0,5. ($\forall i, P_i = 0,5$). Cela signifie que les valeurs des solutions sont prises de manière aléatoire.

A chaque génération, le vecteur P est mis à jour de telle sorte que les nouvelles solutions de la génération qui suivent, auront plus de probabilités de s'approcher vers

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

la meilleure solution de la génération précédente. L'apprentissage est réalisé à chaque fois sur le vecteur P en étant guidé par la meilleure solution.

Algorithme PBIL

Début

Initialiser_vecteur_probabilité (P) /*chaque $P_i = 0.5$ */

Tant que (condition d'arrêt non satisfaite) **faire**

Pour $i=1$ **jusqu'à** $i = N$ **faire** /* N : taille de la population*/

 Solution $_i$ = Générer_Solution_Selon_P(P)

 Evaluation $_i$ = Evaluer(solution $_i$)

Fait

 Max = la meilleure solution

Pour chaque élément P_i **de** P **faire**

$P_i = P_i * (1.0 - LR) + \max_i * LR$

Fait

Pour chaque élément P_i **de** P **faire**

Si (random[0,1] < MUT_P) **alors**

$P_i = P_i * (1.0 - MUT_A) + \text{random}(0 \text{ ou } 1) * (MUT_A)$

FinSi

Fait

Fait

Return Max

Fin

L'algorithme PBIL accepte comme paramètres la taille de la population N qui est aussi le nombre d'individus produit par génération, la probabilité de mutation pour la valeur de chaque position du vecteur de probabilité P (MUT_P), et la valeur de mutation (MUT_A) et enfin le taux d'apprentissage (LR).

La procédure *Générer_solution_Selon_P()* permet de générer une solution en se basant sur la valeur du vecteur P . Le vecteur P et le vecteur produit V possèdent la même taille.

Pseudo-code de Générer_Solution_Selon_P()

```
Début  
Pour i = 1 jusqu'à i= taille(P) faire  
    Si (random[0,1] <Pi) alors  
        V [i]=1  
    Sinon  
        V [i]=0  
    FinSi  
Fait  
Fin
```

Max est le vecteur contenant la solution possédant la meilleure valeur de la fonction fitness (c.à.d. la meilleure solution). Grace au vecteur Max, le vecteur des probabilités P est mis à jour. LR (Learning rate) désigne le taux d'apprentissage, permet de contrôler cette opération d'apprentissage en jouant sur les nouvelles valeurs pour P .

La formule « $P_i = P_i * (1.0 - LR) + \max_i * LR$ » permet à P_i d'augmenter lorsque \max_i est égale à 1, ce qui implique que la probabilité pour que le $i^{\text{ème}}$ gène des prochaines solutions formé à l'aide du nouveau P soit égale à 1 est augmentée. De même si la valeur de \max_i est égale à 0, P_i sera diminué et les nouvelles solutions formées à partir de i auront plus de chance pour que le $i^{\text{ème}}$ gène soit égal à 0. En d'autres termes, les nouvelles solutions générées ont une tendance à se rapprocher vers la meilleure solution des générations suivante.

Une opération de mutation est effectuée aussi au niveau du vecteur P afin de garder toujours un peu de diversité au niveau des solutions obtenus et d'éviter une évolution prématuré. Chaque élément P_i est échangé avec une probabilité MUT_P par la formule suivante : $P_i \leftarrow P_i * (1.0 - MUT_A) + \text{random}(0.0 \text{ or } 1.0) * (MUT_A)$

MUT_A permet de contrôler l'opération de mutation du vecteur P .

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Cet algorithme a été appliqué à des problèmes d'optimisation divers, sauf que rare sont ses applications pour la résolution des problèmes de télécommunication [11]. Dans [25] on a construit un benchmark leurs permettant d'effectuer une large comparaison entre une panoplie d'algorithmes appliqués à l'APP, parmi eux l'algorithme PBIL. Ce dernier a été parmi les algorithmes qui ont fournis les meilleurs résultats.

Remarque: L'algorithme PBIL remplace à chaque itération toute une génération par une nouvelle (ne possède pas de mécanisme d'élitisme).

6.2 Méthodes basées sur la recherche locale

Cette classe de métaheuristiques regroupe les méthodes qui utilisent le principe de la recherche locale. Le problème d'optimisation est résolu par ce type de méthodes de manière itérative. Ceci en faisant à chaque fois évoluer la configuration courante en la remplaçant par une autre configuration de son voisinage.

6.2.1 La recherche locale

Appelée aussi amélioration itérative, la recherche locale représente une ancienne classe de méthodes heuristiques. La recherche locale est fondée sur deux éléments essentiels: un voisinage $N : X \rightarrow 2^{|X|}$ et une procédure P exploitant le voisinage.

Pseudo code de la recherche locale
S= générer solution initial
Répéter
Choisir un voisin S' de S tel que $f(S') < f(S)$
Remplacer S par S'
Jusqu'à ce que pour tout voisin S' de S , $f(S') \geq f(S)$.
Fin

A chaque itération un choix est effectué pour choisir un voisin plus amélioré. Ce choix peut être fait en énumérant les voisins jusqu'à ce qu'on en trouve un qui

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

améliore strictement. Ou bien rechercher le meilleur voisin (meilleure amélioration), malgré que cette dernière s'avère plus coûteuse.

Cette procédure s'arrête lorsque la configuration trouvée ne possède pas de voisins strictement meilleurs qu'elle (optimum local). En conséquence, les solutions trouvées ne sont généralement pas de bonne qualité et sont loin de l'optimum global. Par contre, l'avantage principal de cette méthode consiste dans sa simplicité et sa rapidité.

Des méthodes existent afin de remédier au problème de la recherche locale. En effet, parmi ces méthodes, la *méthode de relance aléatoire* qui consiste à générer aléatoirement une nouvelle solution de départ et recommencer l'opération de nouveau. Par contre cette méthode perd les optimums locaux déjà rencontrés et ne permet pas de les exploiter. D'autres raffinements plus élaborés sont également possibles, par exemple: l'introduction de voisinages variables [19] et les techniques de réduction [22] ou d'élargissement [31].

Définition 4 : Une solution $s \in S$ est un **minimum local** relativement à la structure de voisinage N si $f(s) \leq f(s')$ pour tout $s' \in N(s)$.

Définition 5 : Une solution $s \in S$ est un **minimum global** si $f(s) \leq f(s')$ pour tout $s' \in S$.

Dans la suite de ce chapitre, nous allons présenter les différents types de métaheuristiques basées sur la recherche locale, essentiellement celles qui ont été exploitées pour l'APP.

6.2.2 Algorithme Simulated Annealing (SA)

En français le recuit simulé, Il a été proposé par Kirkpatrick et al [20] L'algorithme SA offre une alternative pour éviter le blocage lors de l'atteinte d'un optimum local. En effet, Au lieu de rester bloqué par ce qu'il n'y a pas de voisin meilleur que la solution obtenue, l'algorithme SA tolère dans des cas similaires de passer vers une autre solution voisine même si elle est moins bonne que la solution précédente (possibilité de dégradation).

Par contre, l'algorithme SA contrôle cette dégradation et est paramétrée en fonction d'un paramètre appelé température. Au fur et à mesure que la recherche avance, moins la dégradation est acceptée.

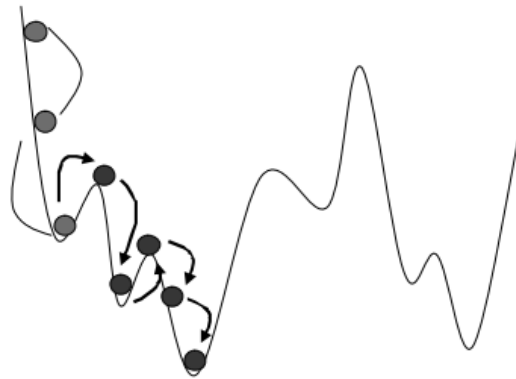


Figure 2.5: Convergence de l'algorithme SA

6.2.2.1 Fonctionnement de SA

L'algorithme SA commence par choisir un point de départ aléatoirement. Ensuite à chaque itération, l'algorithme tente de passer vers un nouvel voisin. Si le voisin possède une meilleure qualité alors il remplace l'ancienne solution, sinon (c.à.d. la nouvelle solution est de mauvaise qualité) cette solution n'est pas rejeté directement, mais elle peut toute fois être acceptée avec une probabilité P .

$$P = 2 / \left(1 + \exp \left(\frac{\text{fitness}(s_a) - \text{fitness}(s_n)}{T} \right) \right)$$

S_a : la solution courante

S_n : la solution voisine

T : le paramètre température

La valeur de T est réduite progressivement à chaque fois qu'un nombre d'itération est effectué, donc P diminue aussi, et moins de solutions seront acceptées à chaque fois que la recherche continu. Cette technique permet d'intensifier la recherche avec le temps. Donc, le paramètre T doit être géré par une fonction de diminution de température.

Algorithme : Recuit simulé SA
Entrée : Une instance du problème à résoudre
Sortie : La solution trouvée
Début

```
t ← 0
Initialiser(T, Sa)
Tanque (condition d'arrêt non satisfaite) Faire
    Tanque (condition de refroidissement de T n'est pas atteinte) Faire
        Sn=choisir un voisin de Sa
        Si (Accept(Sa, Sn, T) Alors
            Sa=Sn
        FinSi
        t ← t+1
    Fait
    Refroidir(T) /*Diminue le paramètre température T*/
Fait
Fin
```

La fonction *Accept* décide si le voisin est accepté, son pseudo code est comme suit :

Algorithme : Accept

```
Entrée : Sa , Sn , T
Sortie : Vrai si Sn est acceptée, faux sinon
Début
Si ( $f(S_a) < f(S_n)$ ) Alors
    Accept= vrai
Sinon
    P= Calcule_P(T)
    Si (random[0,1]< P) Alors
        Accept= vrai
    Sinon
        Accept= faux
    FinSi
FinSi
Fin
```

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

En plus du choix de la fonction objectif (pour calculer la qualité d'une solution) et de voisinage qui doivent être choisies, des paramètres spécifiques à SA doivent aussi être mise au point comme la valeur de la température initial et la fonction de diminution de T .

L'un des principaux défauts de la méthode du Recuit Simulé est le fait de choisir aléatoirement un voisin parmi tous les voisins $N(s)$ possibles, ce qui implique qu'on peut facilement être proche de l'optimum et passer juste à côté sans le trouver.

6.2.2.2 Problème APP résolu avec SA

Cet algorithme a été appliqué sur divers problèmes d'optimisation, parmi eu figure l'APP aussi. Cet algorithme a été utilisé souvent pour les problèmes d'optimisation de télécommunication [21]. Cet algorithme a été utilisé aussi pour comparer de nouveaux algorithmes [3] où il a dépassé les algorithmes gGA et ssGA. Dans [25] il fut comparé avec un large ensemble de méthodes. (Les résultats de ce dernier travail seront résumés dans la dernière section)

Les résultats de l'algorithme SA appliqué au benchmark de [25] n'était pas vraiment remarquable, en fait, il a figuré parmi les algorithmes qui ont été les moins performants et il a été dépassé par d'autres algorithmes comme CHC, PBIL...etc.

6.2.3 La recherche Taboue

La recherche Tabou a été introduite par Glover, F et Laguna, M en 1986 [17] destinée à résoudre les problèmes d'optimisation combinatoire. Elle a été considérée par certains comme une alternative au recuit simulé. Cette méthode est caractérisée par la technique qui permet d'éviter de revisiter les solutions récemment explorées en rendant le déplacement vers ces solutions tabou. Donc ces solutions ne peuvent être visitées tant qu'elles figurent dans une liste appelée liste taboue.

La liste taboue est gérée d'une manière à être mis à jour à chaque déplacement en insérant soit la nouvelle solution soit le mouvement qui a mené vers elle et retirant le plus ancien élément inséré pour devenir non tabou. De plus l'algorithme permet d'échapper aux optima locaux en choisissant toujours de passer vers la meilleure solution non taboue, même si cette dernière est de qualité inférieure.

Il devient généralement un peu couteux en terme de mémoire pour stocker toutes les solutions taboues dans la liste taboue T et en terme de temps pour vérifier si une solution figure dans cette liste, c'est pour cela que généralement c'est le

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

mouvement effectué pour passer d'une solution à l'autre qui est sauvegardé dans cette liste, ensuite ce mouvement devient un mouvement tabou pour une certaine durée de temps.

Par contre cette solution possède quelques lacunes, puisque des fois un même mouvement ne mène pas toujours vers une solution déjà visitée, mais contrairement, des fois ce même mouvement peut mener vers une meilleur solution. L'exemple suivant illustre ce cas : supposant qu'on veut inverser la suite de caractères a,b,c,d, l'algorithme procède comme suite : $abcd \rightarrow bacd \rightarrow dacb \rightarrow dcab$. Les opérations ont été exécutées comme suit : permuté ab, puis bd, et enfin ac. Si $|T| \geq 3$, on n'a donc plus le droit de permuter ab ($|T|$ le nombre d'itération pour qu'une action taboue devienne non taboue) Ainsi, on interdit la solution dcba qui pourrait être la solution optimale. Donc pour remédier à ce défaut, en règle générale, le statut tabou d'une solution est levé si celle-ci est meilleure que la meilleure solution s^* rencontrée jusqu'ici. On appelle ce critère par critère d'aspiration.

Algorithme : La recherche taboue

Entrée : Une instance du problème à résoudre

Sortie : La solution trouvée

Début

Initialiser (S, T, S*)

Tant que (condition d'arrêt non satisfaite) **Faire**

 S' = choisir la meilleure solution de l'ensemble $N^T(S)$

Si ($f(s') > f(s^*)$) **Alors**

 S* = S'

FinSi

 S = S'

 Mettre à jour T

Fait

Fin

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

$N^T(s)$ est l'ensemble des solutions voisines non taboues plus celles qui satisfont le critère d'aspiration.

$$N^T(s) = \{s' \in N(s) \text{ tel que } s' \notin T \text{ ou } f(s') > f(s^*) \}$$

6.3 Algorithme de recherche ILS (Iterated Local Search)

ILS est un algorithme basé sur la recherche locale introduit par Lourenço et al. [24], caractérisé par sa simplicité et par la bonne qualité des résultats rendus. En effet, lorsque appliqué à l'APP dans [25], il a été parmi les algorithmes ayant rendus les meilleurs résultats. L'algorithme ILS effectue une recherche locale sur des solutions modifiées de la solution courante. ILS permet aussi d'échapper à des optima locaux. Par contre, la force principale de cet algorithme vient de son opération d'échantillonnage biaisé pour choisir la première solution aléatoire.

Pseudocode de l'algorithme ILS

Début

$s_0 = \text{genererSolutionInitial}()$

$s = \text{rechercheLocal}(s_0)$

$s_{\text{best}} = s;$

Tant que condition d'arrêt non satisfaite faire

$s' = \text{perturbation}(s, \text{history})$

$s'' = \text{rechercheLocal}(s')$

Si s'' meilleur que s_{best} **alors**

$s_{\text{best}} = s''$

FinSi

$S = \text{critereAcceptation}(S, s'', \text{history})$

Fait ;

Fin

Cet algorithme a été adapté par [25] pour résoudre le problème APP. En effet des heuristiques spécifiques à ce problème ont été intégrées afin de guider la recherche, plus précisément pour créer une solution initiale.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

6.3.1 Génération de la solution initiale

Cet algorithme a été appliqué avec la modélisation décrite dans la section 3.2.3 du chapitre précédent, où les points à couvrir et les transmetteurs sont représentés dans une grille, et la relation entre les transmetteurs et les points qu'ils couvrent est représentée par un hypergraphe.

Pour générer la solution initiale, la grille est décomposée en sous-grilles appelées fenêtres. Ces fenêtres sont toutes de taille $N \times N$, où N est choisi de telle sorte qu'il soit proche du double du diamètre de rayonnement d'une antenne. Ensuite, dans chaque fenêtre, une *BS* est installée au milieu. Une recherche locale est ensuite réalisée à cette solution initiale afin qu'elle soit améliorée plus.

6.3.2 Mécanisme de perturbation

Une opération de perturbation et d'intensification (recherche locale) sont itérativement effectuées jusqu'à satisfaction du critère d'arrêt. Si la solution trouvée dans une itération est meilleure que la première solution, alors celle-ci la remplace.

Le mécanisme de perturbation proposé consiste à choisir quelques transmetteurs utilisés pour les retirer de la solution et d'ajouter un autre ensemble d'endroits où des antennes seront ajoutées. Le nombre de transmetteurs à enlever et celui à ajouter est déterminé aléatoirement par une variable aléatoire qui suit une loi normale de paramètre (μ, σ) , tel que μ est la moyenne et σ est l'écart type. Les transmetteurs à enlever ou à ajouter sont aussi déterminés aléatoirement.

Une dernière étape pour améliorer la solution consiste à calculer pour chaque transmetteur, la fonction objectif dans le cas où il est utilisé et dans le cas aussi où il n'est pas utilisé, et de prendre la meilleure décision comme solution finale.

6.3.3 Mécanisme d'intensification

Une autre opération de recherche locale est effectuée sur la solution obtenue de l'étape précédente, afin de l'améliorer d'avantage.

7 Approches multiobjectif

Jusqu'à-là, ces méthodes présentées sont des méthodes mono objectif, Comme mentionné avant, nous allons résumer dans la section qui suit quelques méthodes et travaux sur les approches multiobjectif.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Une approche multi objectif est caractérisée par l'utilisation d'une méthode de classement d'individus basée sur la définition de la dominance Pareto. Par exemple, Fonseca et Fleming dans [15] a proposé une fonction de classement (*ranking function*) qui assigne à chaque individu un rang égale à $k+1$ s'il est dominé par k individus.

Comme travaux concernant cette approche, on peut citer [27] qui ont implémenté une version de l'algorithme ssGA pour le rendre multiobjectif.

7.1 SSGA

Dans *Osman et al* [27], trois fonctions à optimiser sont considérées, Minimiser le nombre de sites utilisés (f_1), minimiser le taux de trafic perdu (f_2) et minimiser l'interférence (f_3). Toute en respectant deux contraintes: la couverture (les solutions réalisables doivent fournir une couverture minimale de 80%) et le *Handover* (cellules superposées minimum à 50%)

L'algorithme utilisé dans cet article tente de trouver non seulement les sites à activer, mais aussi le type et le paramétrage approprié des antennes utilisées. Donc, le codage hiérarchique (décrit dans la section 3.4 du chapitre précédent) a été employé pour coder les solutions. Les éléments de base de l'algorithme sont dérivés de l'algorithme ssGA (seulement un seul remplacement par génération).

Les opérateurs génétiques (croisement et mutation) ont été modifiés pour correspondre au problème. Le croisement a été modifié pour prendre en compte les informations géographiques (permuté les sites qui sont dans un même rayon géographique donné).

La mutation est modifiée pour toucher à tous les niveaux dans la hiérarchie de la solution, contrairement à la mutation classique où seulement un bit est échangé.

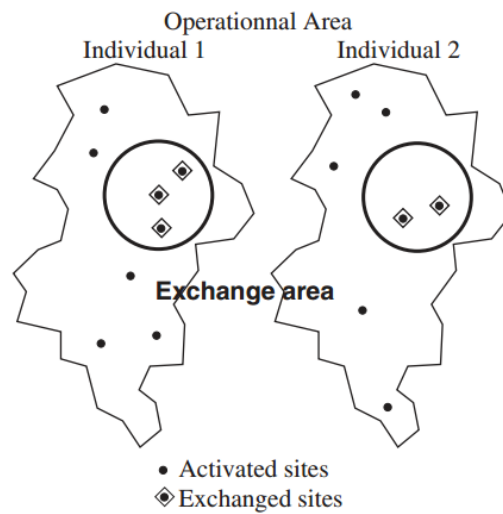


Figure 2.6 : croisement géographique.

Le type de sélection qui a été utilisé est le tirage à la roulette, c.-à-d. qu'un individu S est choisi pour la reproduction suivant une probabilité $P(S)$ tel que cette probabilité est égale à :

$$P(S) = f(S) / \sum_j f(S_j)$$

Ceci veut dire que plus la fonction fitness de l'individu est grande, plus la probabilité de le sélectionner devient grande. Ce qui favorise la reproduction des meilleurs individus.

Le classement des individus est réalisé à l'aide de la fonction de Fonseca et Fleming, qui est basée sur la notion de dominance Pareto. Un individu dominé par k autres individus aura un rang égal à $k+1$.

7.1.1 Partage et élitisme

Le but revient à chercher non seulement des solutions non dominées, mais aussi ces solutions doivent être bien dispersées toutes au long du front Pareto. L'utilisation d'une méthode d'élitisme permet le maintien des meilleures solutions rencontrées, ce qui permet d'intensifier la recherche vers le front Pareto. Dans [27] la technique utilisée consiste à maintenir un archive qui va stocker les solutions Pareto trouvées, où ces dernières vont participer dans la phase de sélection. (Voir figure 2.6). Généralement la plupart des travaux utilisant l'élitisme optent pour la même technique.

Quant à la technique de Partage, elle permet la diversification des solutions obtenues en pénalisant les solutions qui ont une forte densité de solutions voisines.

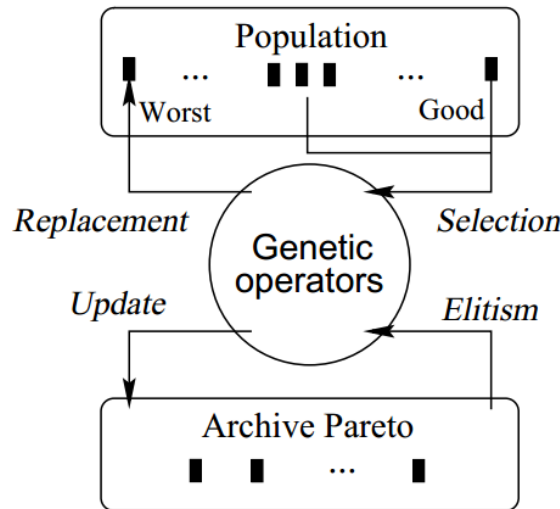


Figure 2.7: Modèle ssGA avec élitisme

Ainsi la fitness d'un individu u redevient :

$$f' = \frac{f(u)}{m(u)}.$$

Tel que $m(u)$ est appelé compteur de niche et est défini par la formule suivante :

$$m(u) = \sum_{v \in \text{population}} sh(\text{dist}(u, v))$$

Où sh est défini par :

$$sh(\text{dist}(u, v)) = \begin{cases} 1 - \left(\frac{\text{dist}(u, v)}{\delta_{sh}}\right)^\alpha & \text{si } \text{dist}(u, v) < \delta_{sh} \\ 0 & \text{Sinon} \end{cases}$$

$dist$ est la distance entre deux individus qui peut être calculer dans l'espace décisionnel ou objectif. δ_{sh} et α sont des paramètres à fixer. Le paramètre δ_{sh} définit la taille de la niche qui signifie la valeur où une solution doit être pénalisée pour chaque autre solution qui est de distance inférieure à cette valeur. La distance $dist$ peut être calculée par la formule suivante

$$\text{dist}(u, v) = \sqrt{\sum_{i=1}^n \frac{f_i(u) - f_i(v)}{f_{max_i} - f_{min_i}}}$$

Cela signifie que plus une solution possède un entourage peuplé d'autres solutions, plus la valeur du compteur de niche est grande. Ce qui amène à pénaliser la valeur de la fitness de cette solution et de rendre moins probable que ce type de

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

solution soit présent dans le front Pareto. Ainsi, le front Pareto obtenu est bien diversifié et ne peut être concentré dans un point donné.

7.1.2 Prise en compte des contraintes

Plusieurs techniques existent pour prendre en compte les contraintes. Une des manières est d'éliminer a priori les solutions qui ne respectent pas ces contraintes. Une autre façon de faire est de pénaliser ces solutions. Cette dernière technique a été exploitée dans cette article, en définissant deux entités $penalty_c$ et $penalty_h$. $penalty_c$ sanctionne les solutions qui ne fournissent pas une couverture totale et fait exclure celles qui fournissent une couverture inférieure à 80%. $penalty_h$ sanctionne les solutions qui ne fournissent pas de prise de main (*handover*) totale et fait exclure celles qui fournissent un *handover* inférieur à 50%.

$$penalty_c(u) = \begin{cases} 5 * \left(\frac{cover}{100}\right) - 4 & \text{si } cover > 80\% \\ 0 & \text{Sinon} \end{cases}$$

$$penalty_h(u) = \begin{cases} 2 * \left(\frac{handover}{100}\right) - 1 & \text{si } handover > 50\% \\ 0 & \text{Sinon} \end{cases}$$

7.2 Algorithme NSGA

Srinvas et Deb (1994) ont mis en œuvre l'idée d'utiliser le concept de *ranking par dominance* dans les AG de Goldberg dans leur méthode NSGA- Nondominated Sorting Genetic Algorithm [18]. Cette méthode joue sur les deux objectifs convergence-diversité, c.à.d. offrir des solutions non-dominées tout en étant diversifiées au long du front Pareto. Ceci par utilisation d'un schéma de calcul de la performance qui favorise les solutions non-dominées et, d'autre part, l'application de la technique de *Partage* entre les solutions du même front non-dominées.

Dans un premier temps, NSGA trie la population P selon le principe de dominance. Elle consiste à diviser la population en un nombre de classes distinctes P_j de façon à ce que tous les individus non-dominés de P appartiennent à l'ensemble P_1 ; ensuite, tous les éléments non-dominés de $P \setminus P_1$ sont placés dans l'ensemble P_2 et ainsi de suite.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Les solutions qui appartiennent à P_1 sont les meilleurs au sens de leur proximité de la surface de Pareto. Ainsi ces individus auront la valeur de performance la plus grande. Cette valeur sera diminuée en passant d'une classe à une autre. Chaque solution i de l'ensemble P_1 reçoit la valeur de performance $f_i = N$.

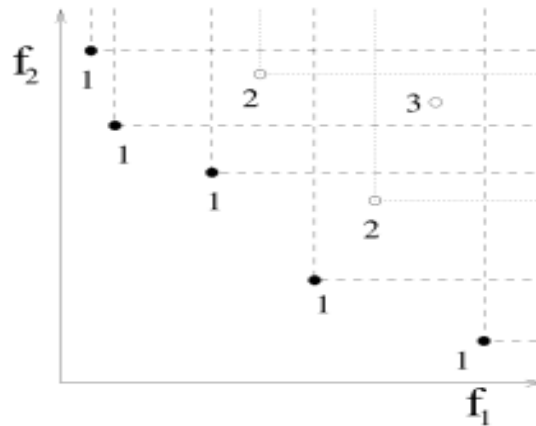


Figure 2.8 : Résultats de *Ranking NSGA*

La prochaine étape permet d'assurer la diversité par la technique du *Partage*. La valeur de performance f d'une solution est dégradée en fonction du nombre de solutions qui se trouvent dans son voisinage (appartenant au même ensemble dominé).

La procédure de partage est la même que celle illustrée déjà dans la section 7.1.1.

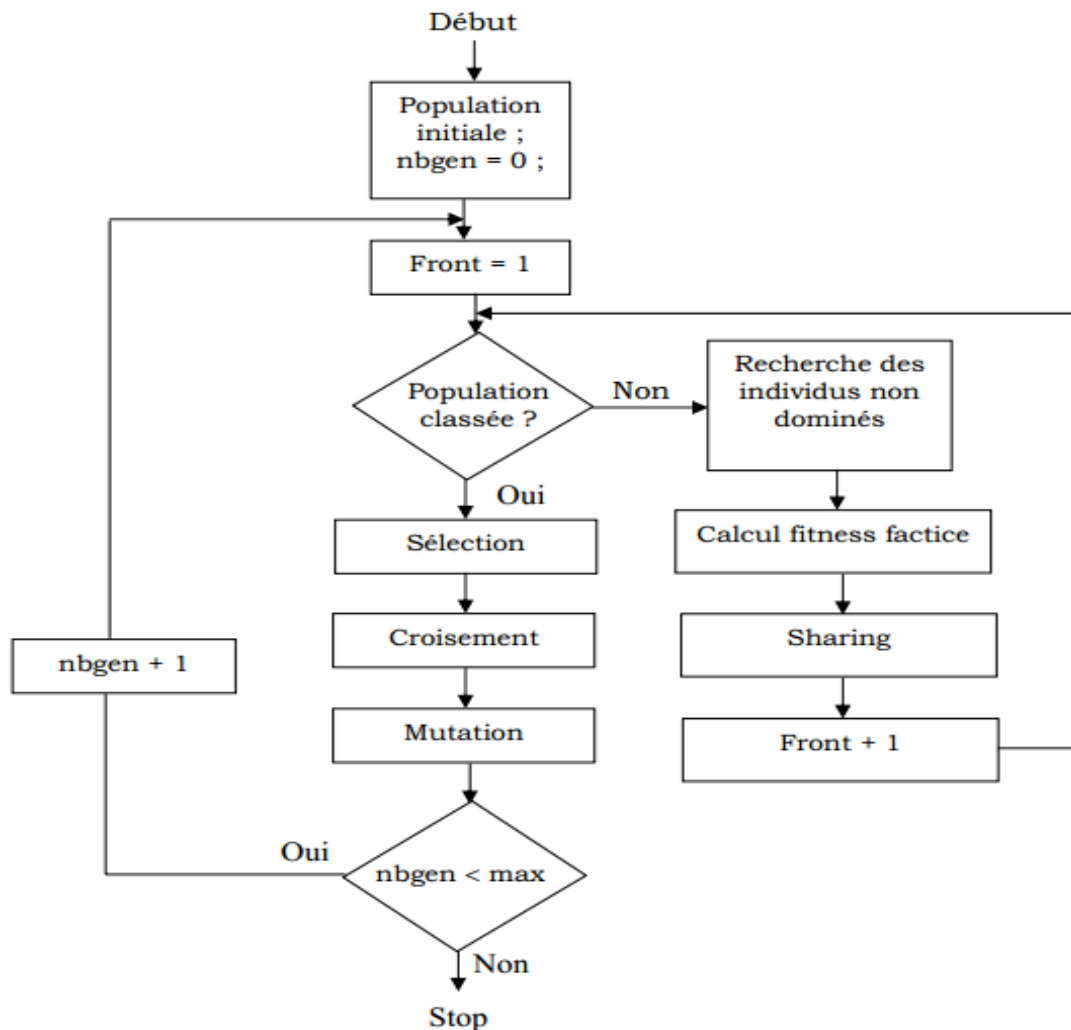


Figure 2.9 : Organigramme de NSGA

Néanmoins, cet algorithme présente quelques insuffisances en raison de sa complexité de calcul et de sa sensibilité au choix de la valeur δ_{sh} (la taille de la niche). Cependant, Deb et al. [13]. ont proposé une nouvelle version pour cet algorithme nommé NSGA-II, qui est plus performante. Utilise l'élitisme et un opérateur de comparaison qui permet de maintenir la diversité sans spécification de paramètres.

7.3 NSGA-II

La méthode précédente ne permet pas de conserver les solutions Pareto-optimales trouvées au cours des générations. Ce genre de méthodes est dit méthodes Non-élitistes. Afin de palier à ce problème, de nouvelles techniques ont été proposées. NSGA-II est l'une des méthodes élitiste. NSGA-II tente de résoudre les problèmes

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

liés à NSGA (non élitiste, complexité de calcul et utilisation de Partage qui implique le réglage de paramètres).

Dans cet algorithme, à chaque génération t une population de parents (P_t) de taille N et une population d'enfants (Q_t) de même taille sont assemblées pour former une population (R_t) de taille $2N$, comme indiqué sur la figure ci-dessous. Cet assemblage permet d'assurer l'élitisme. La population (R_t) est répartie en plusieurs fronts (F_1, F_2, \dots) par une procédure de tri. Une nouvelle population parent (P_{t+1}) est formée en ajoutant les fronts au complet (premier front F_1 , second front F_2 , etc...) tant que ceux-ci ne dépassent pas N . Si le nombre d'individus présents dans (P_{t+1}) est inférieur à N , une procédure de *crowding* est appliquée sur le premier front suivant F_i non inclus dans (P_{t+1}). Le but de cet opérateur est d'insérer les $(N - P_{t+1})$ meilleurs individus de F_i qui manquent dans la population (P_{t+1}). Une fois que les individus de la population (P_{t+1}) sont identifiés, une nouvelle population enfant (Q_{t+1}) est créée par sélection, croisement et mutation. La sélection par tournoi est utilisée mais le critère de sélection est maintenant basé sur l'opérateur de comparaison ($<n$) qui favorise les solutions qui appartiennent aux fronts inférieurs et ceux ayant des distances de *crowding* supérieurs (les solutions qui sont dans des régions dépeuplées) dans le même front. Le processus se répète d'une génération à une autre jusqu'à satisfaction d'un critère d'arrêt.

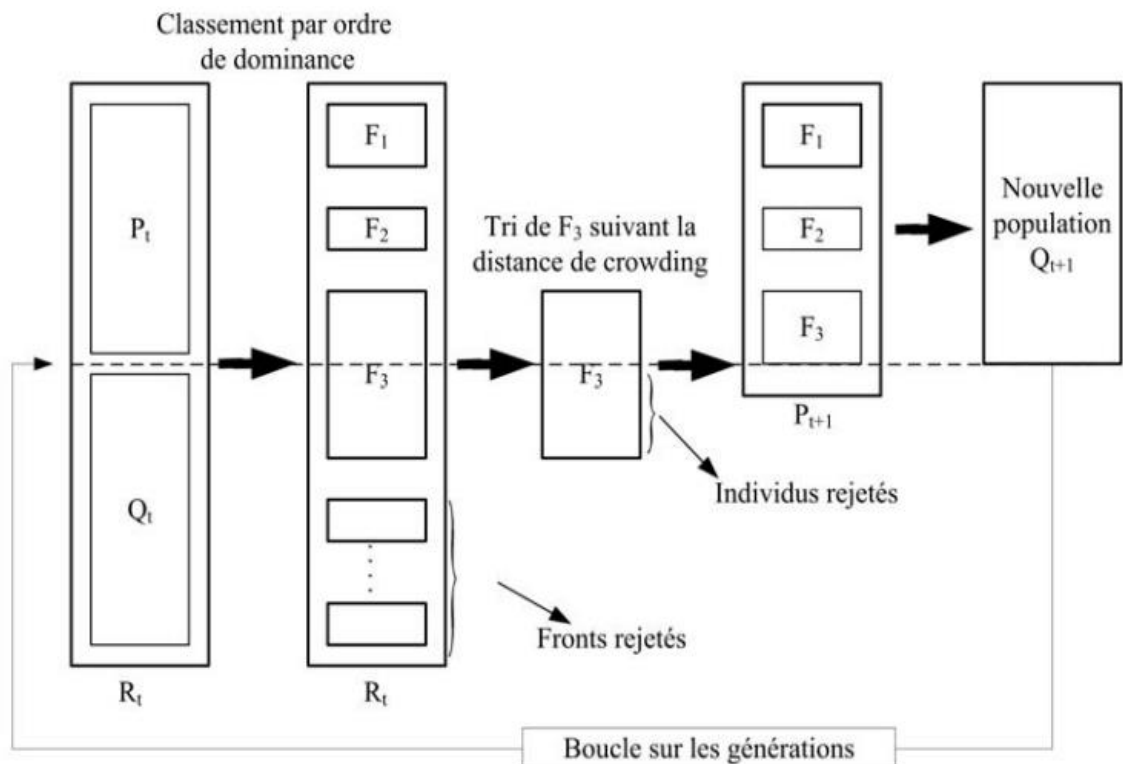


Figure 2.10 : Etapes de l'algorithme NSGAII

7.3.1 Distance de Crowding

Elle remplace la procédure de sharing, la procédure de crowding est basée sur un calcul de distance (distance de crowding) qui ne nécessite aucun paramétrage et d'une complexité algorithmique moindre que celle de sharing. La distance de crowding d'une solution particulière i se calcule en fonction du périmètre de l'hypercube qui a comme sommets les points les plus proches de i sur chaque objectif f . Voici l'algorithme permettant de calculer cette distance

Pseudo code de la procédure de <i>Crowding</i>
<p>Début</p> <p>$N = I$ /* nombre de solutions dans le Front I */</p> <p>Pour chaque solution i faire</p> <p style="padding-left: 40px;">$I[i]_{\text{distance}} = 0$ /* Initialisation des distances */</p> <p>Fait</p>

Pour chaque objectif m faire

$I = \text{trier}(I, m)$ /* trier I par ordre croissant selon le critère m*/

$I[1]_{\text{distance}} = I[N]_{\text{distance}} = +\infty$ /* poser la distance du premier et dernier élément à $+\infty$ */

Pour $i=2$ jusqu'à $|I| - 1$ **faire**

$$I[i]_{\text{distance}} = I[i]_{\text{distance}} + \left(\frac{f_m^{i+1} - f_m^{i-1}}{f_m^{\text{max}} - f_m^{\text{min}}} \right)$$

Fait

Fait

Fin

Pseudo code de NSGAII

Début

Initialiser (P_0, Q_0)

$t = 0$

Tant que ($t < t_{\text{max}}$) **faire**

$R_t = P_t \cup Q_t$

Effectuer un tri selon la dominance de Pareto sur R_t pour identifier les fronts F_j

$P_{t+1} = \emptyset$

$j = 1$

Tant que ($|P_{t+1}| + |F_j| < N$) **faire**

$P_{t+1} = P_{t+1} \cup F_j$

$j = j+1$

Fait

Remplir la population P_{t+1} à partir de F_j en utilisant la distance de *crowding* si $|P_{t+1}| < N$

$i = 1$

$Q_{t+1} = \emptyset$

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Tant que ($i \leq N/2$) **faire**

Sélectionner deux chromosomes C1 et C2 de la population P_{t+1}

Appliquer le croisement et la mutation pour obtenir C1' et C2'

$Q_{t+1} = Q_{t+1} \cup \{C1', C2'\}$

$i = i+1$

Fait

$t = t+1$

Fait

Fin

Le grand avantage de NSGAI est l'utilisation d'une stratégie de préservation de diversité qui n'exige aucun paramètre à fixer. Par contre, NSGAI effectue un classement sur une population de $N \times 2$ individus au lieu de N , ce qui constitue un des inconvénients de cette méthode.

7.4 SPEA 2

C'est une version améliorée de l'algorithme SPEA de Zitzler et Thiele (1998) [33] qui introduit l'élitisme par le maintien explicite d'une population externe P' qui contient un nombre limité de solutions non-dominées trouvées depuis le début de l'évolution.

Malgré que SPEA permet de garder les solutions Pareto-optimale trouvées et qu'il possède un principe de calcul de performance scalaire assez simple, ce dernier possède quelques inconvénients. On cite par exemple la dépendance de la performance de l'algorithme au réglage de la taille du registre par rapport à la taille de la population courante. Aussi SPEA utilise un mécanisme de (regroupement) clustering pour garder la taille de l'archive fixe qui tend à ne pas préserver les individus qui se trouvent sur les extrémités de l'archive.

Pour cela SPEA 2 a été introduit par les mêmes auteurs de SPEA [34]. Ce dernier diffère de son prédécesseur dans plusieurs aspects.

- La taille de l'archive est fixe.
- Un système d'affectation de performance amélioré, qui prend en compte pour chaque individu le nombre d'individus qui le dominent et le nombre de ceux qu'il domine.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

- De nouvelles méthodes de troncature d'archive qui garantissent la préservation des solutions qui se trouvent dans l'extrémité de l'archive.

Dans [34], une étude comparative a montré que SPEA-II fournit de bonnes performances en termes de convergence et de diversité des solutions. Il a été montré aussi que ses performances dépassent ceux de la première version (SPEA), et se compare bien à d'autres algorithmes comme NSGA-II et PESA.

7.4.1 Affectation de performance

Avec SPEA-II pour chaque individu deux informations sont prises en compte (les solutions dominées et les solutions dominantes) pour calculer la valeur de la performance.

Dans un premier temps, on calcule pour toutes solutions une valeur appelée *strength* S_i .

$S_i = |\{j/j \in P_t \cup P'_t \wedge i \succ j\}|$ tel que P_t désigne la population courante et P'_t la population externe (l'archive) et \succ correspond à la relation de dominance.

Sur la base des valeurs S , la performance préliminaire R_i (raw fitness) est définie comme suite :

$$R_i = \sum_{j \in P_t \cup P'_t, j \succ i} S_j.$$

Il est important de souligner que les valeurs de performances sont à minimiser. Une valeur grande de R_i signifie que l'individu i est dominé par plusieurs autres individus qui à leurs tours dominent plusieurs individus. Cette manière d'attribution de performance offre un mécanisme de garder les solutions de la population diversifiées, par contre ce mécanisme ne s'avère pas efficace dans le cas où la plupart des individus ne se dominent pas les uns les autres. Pour cela, une autre technique est utilisée pour estimer la densité des solutions afin de distinguer entre les individus ayant des valeurs de performances unique.

La technique d'estimation de la densité utilisée est une adaptation du "k^{ème} plus proche voisin" [30] où la densité à chaque point est une fonction décroissante de la distance par rapport au k^{ème} plus proche voisin. Ici, l'inverse de la distance par rapport au k^{ème} plus proche voisin comme l'estimation de la densité. Ici, pour chaque individu i on calcule les distances par rapport à tous les individus j dans l'espace des objectifs. Ces distances sont triées par ordre croissant et le k^{ème} élément donne la

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

distance notée σ_i^k . prendre $k = 1$ est suffisant et mène à une implémentation efficace. Ainsi la densité $D(i)$ est donnée par la formule suivante :

$$D(i) = \frac{1}{\sigma_i^k + 2}$$

On ajoute la valeur 2 au dénominateur afin d'éviter qu'il soit nulle et que la densité soit inférieure à 1. Ainsi la valeur de performance définitive est donné par :

$$F(i) = R(i) + D(i)$$

Dans l'étape de la mise à jour de l'archive \mathbf{p}'_t , les valeurs de performances sont utilisées. Les individus ayant des valeurs de F inférieure à 1 (non dominés) sont copiés dans l'archive de génération suivante.

$$\mathbf{p}'_{t+1} = \{i \mid i \in P_t \cup \mathbf{P}'_t \wedge F(i) < 1\}$$

Si la taille de l'archive est inférieure à N' (la taille fixe de l'archive), celui la est complété par les meilleurs individus dominés (classés par rapport à F).

Dans le cas où le nombre d'individus non dominés dépasse N' , une procédure de troncature de l'archive est invoqué qui élimine les individus qui se trouvent à la plus petite distance d'un autre individu de \mathbf{p}'_{t+1} .

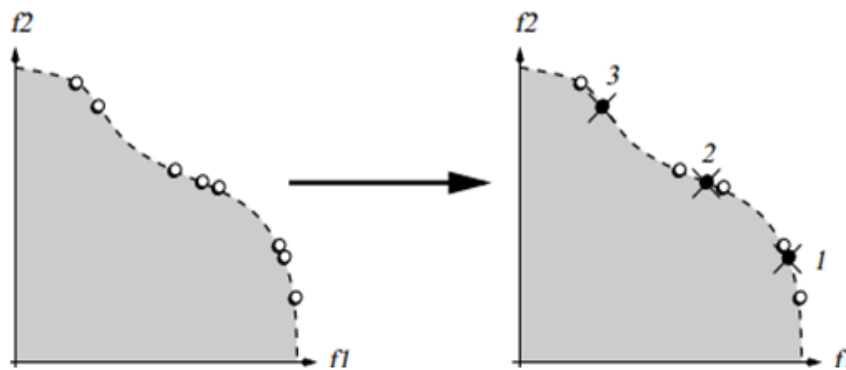


Figure2.11 : procédure de troncature

Dans les travaux d'optimisation sur l'APP généralement, les concepteurs de nouvelles méthodes font référence aux algorithmes NSGAI et SPEAI pour comparer leurs résultats obtenus. Comme dans [6] avec l'algorithme MOCHC. Et dans [36] avec l'algorithme IBEA, qui a été proposé dans [35]. Ce dernier peut être adapté aux préférences de l'utilisateur et plus encore, il ne nécessite aucun mécanisme de préservation de la diversité.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

7.5 Algorithmes MOCHC

MOCHC est une extension de l'algorithme CHC qui a été proposé par Eshelman en 1991 [14]. Basé sur CHC, l'algorithme MOCHC a été conçu pour s'adapter aux problèmes multiobjectif.

Dans [6] *Alba et al* ont remarqué la bonne performance de l'algorithme CHC et des résultats fournis par ce dernier. Ils ont essayé de tirer profit de cet algorithme et de voir si une version modifiée de celui-là de tel sorte à le rendre multiobjectif de voir s'il offre de meilleures performances encore.

Le changement opéré au niveau de CHC a été au niveau du mécanisme de sélection élitiste la fonction d'évaluation. Le classement des solutions par un nombre scalaire effectué par l'algorithme CHC a été remplacé par un mécanisme de *ranking* et de l'estimateur de distance de *crowding* comme celui utilisé dans NSGA II (voir section 7.3).

L'algorithme MOCHC possède aussi un mécanisme de redémarrage, lancé lorsque la population arrive au point mort, C.à.d. l'évolution ne génère plus des solutions de meilleure qualité que celle déjà présente. Ce mécanisme est une forte opération de mutation effectuée sur toutes les solutions sauf celles de bonne qualité. En d'autres termes, garder un pourcentage de solutions après les avoir classifiées par le *ranking* et la distance de *crowding*. Ainsi ce pourcentage de solution à garder sans les faire passer par le mécanisme de redémarrage est un paramètre de l'algorithme MOCHC. Dans ce même article, où il a été appliqué à l'APP ils ont proposé de garder 5% de solutions.

7.5.1 Prise en charge des contraintes

Les contraintes sont considérées aussi par la même méthode que celle utilisée dans NSGAI. Lorsque deux solutions sont comparées, celle qui transgresse le moins de contraintes est la solution préférée.

7.5.2 Performances MOCHC

Les résultats obtenus par des expérimentations faites sur une instance du réseau modélisée par la méthode de [10] (voir section 3.2) où la solution est préalablement connue, les résultats ont été nettement améliorés utilisant MOCHC par rapport à ceux de CHC. En effet, la performance de MOCHC dépasse celle de CHC d'environ 50% en ce qui concerne le nombre d'itération nécessaire pour atteindre une solution

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

optimale. MOCHC a été aussi plus performant lorsque comparé à d'autres algorithmes (NSGAI et SA) par rapport à la même mesure de performance (nombre d'itérations nécessaire pour atteindre la solution optimale). La qualité du front Pareto rendu par NSGAI et MOCHC a été comparée aussi. Et la aussi MOCHC a rendu un front Pareto de meilleur qualité.

8 Approches parallèles

La haute complexité du problème et la charge de calcul importante due à la nécessité d'avoir une population d'une assez grande taille, afin que les algorithmes génétiques aboutissent à de bonnes solutions a conduit à une tendance qui essaye de partager le traitement entre plusieurs processus, et de penser aussi à un autre paradigme qui consiste à rendre ces algorithmes parallèle. En fait, ceci est aussi justifier aussi par l'existence de tâches indépendante (Calcul de la valeur fitness, mutation, etc).

Un des problèmes aussi des algorithmes génétique, est la convergence rapide (prématuré) ce qui cause des résultats qui ont des valeurs de la fonction fitness pas tellement élevées. Comme solution pour ce problème est de décomposer la population en plusieurs sous populations appelées (îles) [32] qui évolues chacune indépendamment de l'autre et qui coopères entre eux, par un mécanisme de migration permettant à une population de profiter des bonnes résultats trouvés dans les autres populations. La migration consiste à faire migrer les meilleures solutions trouvées d'une population à une autre dans un ordre prédéfini.

Toutefois, la structuration de la population n'est pas la seule technique pour implémenter les algorithmes parallèles, une autre méthode consiste à rendre le calcul des opérations indépendante parallèle, sans restructurer la population. Comme dans le cas de *Global parallelism approach* [23] où le modèle est un modèle séquentiel mais l'implémentation est parallèle (le calcul de la fonction fitness, croisement, mutation...).

8.1 Modèle proposé par Caligari

Beaucoup de travaux appliqués sur l'APP se sont consacrés à ce type d'approche et ont proposés chacun son modèle, comme par exemple [10] et [4].

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Dans *Calegari et al*, [10], on a montré comment un algorithme génétique est rendu parallèle en se basant sur le concept d'îles. L'algorithme proposé fut appelé IPGA (*Island-based Parallel Genetic Algorithm*).

L'algorithme a été implémenté sur un réseau, tel que chaque île sera pris en charge par un poste. La direction des communications entre îles forme un anneau virtuel orienté.

A chaque fois qu'une nouvelle génération est calculée, une copie du meilleur individu qui a la plus grande valeur fitness jamais rencontré par chacune des îles est envoyée à la prochaine île. Ce dernier individu prend aléatoirement la place d'un autre individu de cette île.

Plusieurs configurations ont été testées sur ce modèle ont permis de faire quelques constatations. Le teste a montré que les solutions obtenues par un algorithme à 4 îles de 40 individus chacun, sont meilleures que ceux obtenues par un simple GA avec une population de 160 individus, et que ceux obtenues par un algorithme à 40 îles de 4 individus sont encore meilleures que les deux premiers

L'auteur conclu que le nombre d'îles influence sur la qualité des solutions rendues. Une explication possible est que la migration des individus apporte une diversification à la population qui accueille ces derniers. Ce qui empêche la convergence prématuré des populations donc de meilleurs résultats sont obtenus.

La figure 2.11 montre que dans le cas où 10 îles sont utilisées, l'augmentation de la valeur fitness est plus rapide et plus régulière que dans le cas où aucune île n'est utilisée.

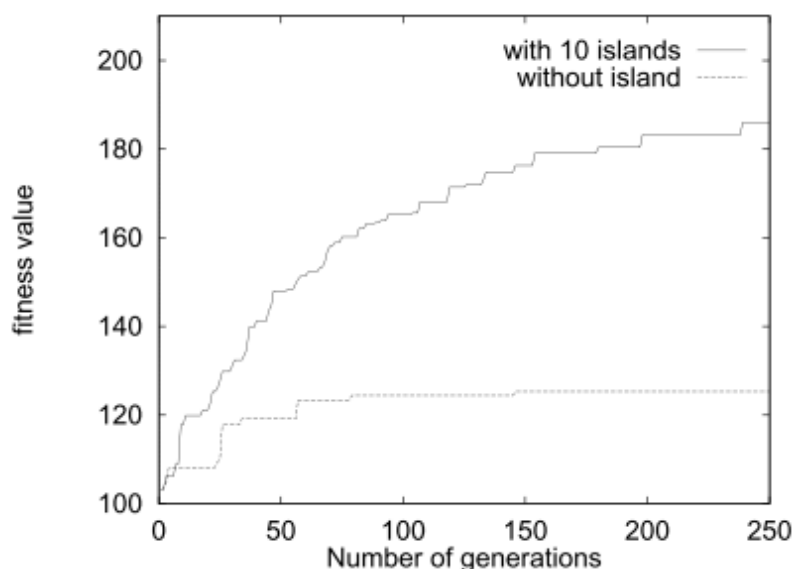


Figure 2.12 : Evolution de la valeur fitness sur une seule population à 160 individus et sur 10 îles à 16 individus chacune.

Une autre série de tests a permis de montrer que la vitesse de l'algorithme augmente presque linéairement avec le nombre de postes utilisés. L'algorithme utilisé comporte 80 îles de 2 individus chacune. Ces résultats ont aussi été confirmés lorsque des expérimentations ont été appliquées sur une instance réelle du problème.

8.2 Autres travaux sur les approches parallèles

Dans Alba et al [4] on a aussi proposé un algorithme parallèle qui l'a nommé dssGA (*distributed steady stateGA*), qui est une tentative de rendre l'algorithme ssGA parallèle. La même technique basée sur les îles a été exploitée.

Deux configurations de dssGA (avec 8 îles) et ssGA ont été comparées. dssGA a été implémenté sur un seul processeur comme première configuration et sur 8 processeurs dans la deuxième configuration.

Malgré que numériquement, le nombre d'évaluation avant d'atteindre l'optimum, ssGA agit mieux que les deux autres algorithmes, mais en terme de temps d'exécution dssGA exécuté sur 8 processeurs consomme moins. Ça s'explique par la grande capacité calculatoire offerte par les 8 processeurs.

D'autres travaux ont porté aussi sur les techniques de parallélisations des algorithmes évolutionnaires comme par exemple [16].

9 Conclusion

Nous avons présenté dans ce chapitre les différentes approches métaheuristiques qui ont été utilisées comme moyen de résolution au problème de positionnement d'antennes dans les réseaux GSM, à savoir les approches de résolution mono et multi objectif. Pour ces deux approches, nous avons présenté des algorithmes évolutionnaires ainsi que des algorithmes de trajectoires.

Nous avons vu aussi que la charge de travail importante des approches évolutionnaire a donné naissance au nouvel paradigme qui s'agit de rendre ces algorithmes parallèles, ce qui nous a mené à faire une petite introduction sur ce concept.

Nous avons vu aussi que même si les approches de trajectoire nécessitent moins de charge de travail, elles rendent généralement de bon résultats. Ce qui nous a mener à s'intéresser à ces méthodes dans notre travail.

Dans le chapitre suivant nous présenterons nos approches pour la résolution du problème de positionnement d'antennes dans les réseaux GSM.

Chapitre 3 :

Approche proposée

1 Introduction

Dans ce chapitre, nous traiterons trois algorithmes : l'algorithme ILS, l'algorithme BLS et notre algorithme inspiré de l'algorithme ILS. L'algorithme BLS qui est un nouveau algorithme aussi inspiré de l'algorithme ILS et qui a été utilisé pour des problèmes autres que le problème de positionnement d'antennes sera adapté pour la résolution de ce problème. Nous présenterons aussi notre algorithme proposé.

La variante du problème résolu sera aussi décrite, c.à.d. on illustrera les différents contraintes du problème (déjà définies dans un chapitre antérieur) qui seront prises en compte.

2 Délimitation du problème résolu

Évaluer une solution pour un problème APP qui englobe toutes les objectifs et caractéristiques à la fois est une tâche très couteuse, ce qui oblige généralement à simplifier le problème et à ne traiter que certains objectifs et contraintes à la fois. De plus, opter pour une modélisation du problème qui est très proche de la réalité ne permet pas de se focaliser sur l'efficacité de l'algorithme. Pour ces raisons, nous n'allons considérer comme objectifs que la maximisation du taux de couverture de la surface et la minimisation du nombre d'antennes utilisées.

3 Modélisation

La modélisation de Mendes décrites dans le chapitre précédent convient parfaitement pour notre cas. De plus, Dans leurs articles [25] ils proposent un benchmark où plusieurs algorithmes ont été comparés. Nos résultats pourront être comparés avec ceux présentés dans cet article.

3.1 La zone géographique

La zone géographique à couvrir est discrétisé, c.à.d. elle est représentée comme un ensemble de points fini. Chaque point est caractérisé par ses coordonnées géographique (x,y) . Il est à noter que les obstacles ne sont pas traités et que la zone est considérée comme plate qu'on appelle zone homogène. Mais en cas où la zone

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

contient des endroits difficiles et qui ne sont pas habités comme la mer ou les montagnes, ces endroits sont traités par la non installation de BSs sur eux.

3.1.1 Matrice

Une matrice suffit pour représenter notre zone. Les éléments de cette matrice seront les points géographiques où leurs positions géographiques seront leurs positions dans la matrice.

3.1.2 Point

Un point de position (x ,y) donnée possède plusieurs caractéristiques :

Le type : Deux types de points sont possible, le type L pour un point normal et le type M pour un point qui contient une BS.

La couverture : indique si le point est couvert ou non.

Les BS voisines : stock les ids des BS voisines. Si une BS est activée le point est automatiquement considéré comme couvert.

Id de BS : Id de la station de base qui est installée sur ce point (seulement dans le cas où le point est de type M)

3.2 Antennes

Nous avons opté pour une antenne de type isotrope pour raison d'alléger le problème là aussi. En effet, le processus de définition de cellules est beaucoup plus simple lorsqu'un modèle de propagation d'ondes isotrope est utilisé au lieu d'un modèle réel.

La forme d'une cellule est circulaire lorsque le modèle isotrope est utilisé, car une antenne isotrope propage le signal de façon uniforme dans toutes les directions.

Donc pour une BS donnée qui est activée, tous les points qui sont de distance inférieure au rayon de couverture de l'antenne sont considérés comme couverts.

La distance entre un point $P_M (x_{P_M}, y_{P_M})$ (contenant une station de base) et un point $P_i (x_{P_i}, y_{P_i})$ de type quelconque est donnée par la formule suivante :

$$\text{Dist} (P_M , P_i) = \sqrt{(x_{P_M} - x_{P_i})^2 + (y_{P_M} - y_i)^2}$$

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Afin de gagner en terme de temps de calcul ces distances seront calculées avant le lancement de notre algorithme. Donc chaque BS possèdera sa liste de Points voisins. Ainsi, on n'aura pas besoin de calculer ces distances pour chaque évaluation d'une solution, mais on aura qu'à parcourir ces listes lorsque l'on veut mettre à jour l'état de ces points.

4 Fonction objectif utilisée

Nous utiliserons la fonction objectif déjà décrite dans le chapitre précédant.

$$f(s) = \frac{\text{taux de couverture}^\alpha}{\text{nombre de BS utilisé}}$$

Nous prenons ici $\alpha=2$.

5 Evaluation d'une solution

Afin d'évaluer une solution donnée, nous avons besoin de déterminer son taux de couverture sur la zone. Pour ça, nous devons tout d'abord distinguer les points qui sont couverts par cette solution.

5.1 Algorithme de couverture

L'algorithme de couverture parcourt le vecteur solution et pour chaque élément i qui est mit à 1 (BS_i activée) on met tous les points se trouvant dans un diamètre inférieur à celui de couverture de l'antenne à couverts.

Algorithme de couverture
Entrée : La zone à couvrir
Sortie : Points couverts
Paramètre :
Diamètre : le diamètre de couverture de l'antenne
Début
/* S : vecteur solution où chaque élément i indique si BS_i est activée*/

```
Pour chaque élément  $S[i]$  faire

  Si ( $S[i] == 1$ ) alors
    Pour j = -
      Diamètre à j = + Diamètre faire
        Pour k = - Diamètre à k = + Diamètre faire
          Si ( $\text{Dist}(BS_i, P_{jk}) < \text{Diamètre}$ ) alors
            Mettre  $P_{jk}$  à couvert
          Finsi
        Fait
      Finsi
    Fait
  Finsi
Fait ;
```

5.2 Algorithme d'évaluation de la solution

Voici l'algorithme qui permet d'évaluer une solution donnée.

Algorithme d'évaluation de la solution

Entrée : une solution à évaluer

Sortie : la valeur de la fonction objectif de la solution d'entrée

Paramètre :

NbP : nombre de points contenus par la zone

Début

/* distinguer les points couverts*/

Couvrir_zone(S)

/*Compter le nombre de points couverts*/

Pour chaque P_i de la zone faire

Si (couvert(P_i) == vrai) **alors**

 NbPC=NbPC+1

FinSi.

```
Fait ;  
Cover_Rate= NbPC / NbP  
  
/*Compter le nombre de BS activée*/  
Pour chaque S[i] faire  
    Si (S[i] == 1) alors  
        NbBS =NbBS+1  
    FinSi.  
Fait;  
/*Calculer la valeur de f(S)*/  
f(S) = Cover_Rate2 / NbBS  
Fin
```

5.3 Voisinage

Les algorithmes considérés dans notre travail sont basés sur la recherche locale, ce qui veut dire qu'ils cherchent l'amélioration d'une solution en explorant son voisinage. Donc un voisinage d'une solution doit être défini.

Afin d'obtenir l'ensemble des voisins d'une solution, nous avons défini deux mouvements, un mouvement aléatoire M1 et mouvement dirigé M2.

Le mouvement aléatoire modifie de manière aléatoire des éléments du vecteur. Ces éléments sont choisis aléatoirement.

Le mouvement dirigé prend en considération des informations liées au problème. En effet, au lieu d'avancer avec des mouvements où la valeur des éléments du vecteur est changée aléatoirement, ici on choisit pour un mouvement une BS à activer et on désactive les autres BS qui sont dans un voisinage de rayon R qui est un paramètre à fixer empiriquement. Cette idée nous permet d'éviter de passer par beaucoup de solutions qui ont des BSs activées trop proches les uns des autres, car on sait qu'une bonne solution ne possède pas ces BSs. En effet, deux BS trop proches d'elle-même permettent de couvrir le minimum de surface que peut couvrir deux BS qui sont bien positionnées. On dit que la recherche est dirigée par l'information qui concerne le positionnement des BS ainsi que leur état (activée ou non).

Notant aussi que ce mouvement dirigé élimine le besoin d'une liste de mouvement Tabou exigé par l'algorithme BLS, car on sait que le rôle de la liste tabou

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

est de prévenir le retour vers un même état pendant un nombre de mouvement défini, alors qu'on sait aussi que dès que le mouvement dirigé qui a été défini soit effectué, on ne peut revenir à l'état précédente par un petit nombre de mouvements. Ceci nous permet de gagner aussi en termes de diminution de la complexité du problème car on élimine les traitements de gestion de la liste Tabou (mise à jour, vérification) pour chaque mouvement effectué.

6 Algorithme ILS

Nous avons auparavant montré le fonctionnement générale de l'algorithme ILS dans le chapitre précédent, par contre nous allons dans cette section décrire les points nécessaires pour son adaptation à notre solution. L'implémentation d'ILS nous permettra de comparer les résultats obtenus avec notre approche BLS.

Comme nous l'avons souligné déjà, l'algorithme ILS utilise principalement un mécanisme de perturbation pour échapper aux optimum locaux. Par contre ce mécanisme de perturbation n'est contrôlé que par le paramètre «*strength*» qui gère la force de perturbation de la solution.

ILS utilise aussi un mécanisme de prévention des cycles de recherche redondant (le retour continue vers le même optimum local), par emploi d'un compteur qui compte le nombre d'itération où la meilleure solution trouvée n'a pas été améliorée. Si le nombre d'itération effectué sans amélioration dépasse un certain paramètre b , la valeur de la variable *strength* est augmentée de 1. (initialement cette variable est égale à 1). Suivant la valeur de *strength*, l'intensité de la perturbation est définie, en effet, *strength* définit en réalité le nombre de case touchées par la perturbation dans le vecteur solution.

Algorithme ILS
Entrée : Une instance du problème, condition d'arrêt
Sortie : Meilleure solution S_{best}
Début
$s_0 = \text{genererSolutionInitial}()$

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

```
s = rechercheLocal(s0)
```

```
Sbest = s;
```

```
Tant que condition d'arrêt non satisfaite faire
```

```
  s' = perturbation (s, history)
```

```
  s'' = rechercheLocal(s')
```

```
  Si s'' meilleur que Sbest alors
```

```
    Sbest = s''
```

```
  FinSi
```

```
  S = critèreAcceptation(S, S'', history)
```

```
Fait ;
```

```
Fin
```

6.1 Mécanisme de perturbation d'ILS

ILS emploie un seul mécanisme de perturbation contrôlé par la variable *strength*. Nous définissons donc le processus de perturbation comme suit : on choisit aléatoirement parmi les BS activées celles qui seront désactivées et parallèlement, on choisit parmi les BS désactivées celles qui seront activées. Le nombre de BS modifiées est égal à la valeur *strength*.

6.2 Recherche local employée par ILS

Afin d'éviter que la recherche soit trop lente, nous avons décidé que lors de l'exploration du voisinage par la recherche local, la solution actuelle est remplacée par le premier voisin de meilleure qualité rencontré. Ceci nous évitera de parcourir tout le voisinage pour chaque solution, même si des fois le parcours de tous le voisinage permet d'obtenir des voisins meilleurs.

Cette stratégie adoptée s'avère plus efficace lors que la recherche explore de mauvais voisinages, car ça nous permet de passer plus rapidement vers un autre voisinage dès qu'on rencontre une solution améliorante au lieu de rester dans le même voisinage pour l'explorer complètement. La recherche est arrêtée lorsqu'aucune meilleure solution n'est trouvée.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Algorithme de la recherche locale

Entrée : une solution donnée

Sortie : optimum local

Début

Déterminer voisin de S_c

Choisir une solution voisine meilleure que S_c

$S = \text{voisin}(S_c)$

Tant que ($S \neq S_c$) faire

$S_c = S$

 Déterminer voisins de S_c

 Choisir une solution voisine meilleure que S_c

$S = \text{voisin}(S_c)$

Fait ;

Fin

6.3 Algorithme de perturbation employé par ILS

Algorithme PerturberSolution

Entrée : solution, L : degré du saut, nombre de fois où la solution optimal n'a pas été améliorée

Sortie : solution perturbée

Début

Pour $i=0$ jusqu'à $i = strength$ **faire**

 Choisir une BS activée aléatoirement et inverser son état

 Choisir une BS désactivée aléatoirement et inverser son état

 Choisir une BS aléatoirement et inverser son état

Fait ;

Fin

6.4 Critère d'acceptation

À partir d'un optimum local S^* ILS effectue premièrement une perturbation ensuite une recherche local pour en arriver à un autre optimum local S'^* , ILS relance

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

la procédure (perturbation-recherche local) à partir d'une solution choisit entre S^* et $S^{*'}$. Le critère d'acceptation est une fonction qui permet de choisir entre S^* et $S^{*'}$ comme un autre point de départ pour la recherche.

Cette fonction peut être de telle manière à choisir toujours la meilleure des deux solutions.

$$\text{Better}(S^*, S^{*'}, \text{history}) = \begin{cases} S^{*'} & \text{si } S^{*' \text{ est meilleure que } S^* \\ S^* & \text{sinon} \end{cases}$$

Cette solution favorise plus l'intensification de la recherche. Par contre dans le cas opposé le critère d'acceptation peut être une fonction appelée random walk noté RW. C'est-à-dire en choisie toujours le nouveau optimum quel que soit sa qualité.

$$RW(S^*, S^{*'}, \text{history}) = S^{*'}$$

Cette fonction favorise plus la diversification au lieu de l'intensification.

7 Algorithme BLS

BLS a été utilisé auparavant pour la résolution de plusieurs problèmes liés aux graphes, parmi eux : Problème de la coupe maximum [8] dans les graphes et Problème de la somme coloration d'un graphe [9].

Notre travail sera donc de définir l'algorithme BLS ainsi que les changements apportés afin d'adapter cet algorithme au problème APP.

7.1 Fonctionnement de BLS

L'algorithme BLS alterne entre deux phases, la phase recherche locale qui essaye de trouver l'optimum local et la phase perturbation pour relancer la recherche à partir d'une nouvelle solution obtenue à l'aide du dernier optimum. Cette dernière phase permet à l'algorithme de s'échapper de l'optimum local et d'étendre la recherche sur d'autres zones.

7.1.1 La recherche locale dans BLS

La recherche locale employée par BLS est la même que celle d'ILS décrite en haut.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

7.1.2 Mécanisme de perturbation de BLS

Le mécanisme de perturbation que possède BLS constitue sa partie principale. Grâce à ce mécanisme BLS échappe aux optimums locaux et permet de visiter plusieurs voisinages. Ainsi, l'espace de recherche est plus varié, ce qui permet de visiter plusieurs solutions.

BLS emploie une sorte de perturbation adaptable selon l'état de la recherche. Pour ça, il possède deux types de perturbations : perturbation dirigée (Faible perturbation) et perturbation non dirigée (Forte perturbation). Le premier type de perturbation est employé lors du début de la recherche et à chaque fois qu'il y'a une amélioration de S_{best} (la meilleure solution trouvée depuis le début de la recherche). Par contre le second type est employé lorsque la recherche est stagnée et n'arrive plus à améliorer S_{best} .

Le choix d'un type de perturbation à effectuer dépend du paramètre T . Par contre le degré de perturbation lui dépend d'un autre paramètre appelé *Degré du saut* (nombre de mouvement à réaliser).

Dans un premier temps, BLS emploie une faible perturbation avec un *Degré du saut* égale à une valeur minimale. Si BLS parcourt consécutivement le même optimum local alors le paramètre *Degré du saut* est augmenté à chaque fois. Ceci permet d'augmenter le degré de perturbation pour changer la zone de recherche qui semble converger toujours vers le même optimum. A chaque optimum local trouvé, BLS le compare avec la meilleure solution trouvée jusque-là. Si BLS parcourt un nombre d'optimum supérieur au paramètre T sans aucune amélioration, BLS considère la zone de recherche actuelle comme infructueuse et passe à une nouvelle zone grâce à une forte perturbation.

Dans notre cas, ces deux types de perturbation sont réalisés comme suite :

7.1.2.1 Perturbation dirigée

Ce type de perturbation est dirigé par l'utilisation de la liste tabou pour n'effectuer que les mouvements non tabous. Il est dirigé aussi par l'utilisation de mouvements qui sont basés sur des données liées au problème.

Nous avons défini deux mouvements pour ce type de perturbation :

P1 : Choisir un point non couvert et activer une BS non taboue dans son voisinage.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

P2 : choisir une BS non taboue aléatoirement et lui appliquer un mouvement de type M2 (voir la section 5.3).

7.1.2.2 Perturbation non dirigée

On active quelques BS et on désactive quelques autres aléatoirement. Le nombre de Bs allumées et éteintes est égal au paramètre *degré du saut*. La nouvelle solution obtenue sera le prochain point de départ pour la recherche locale.

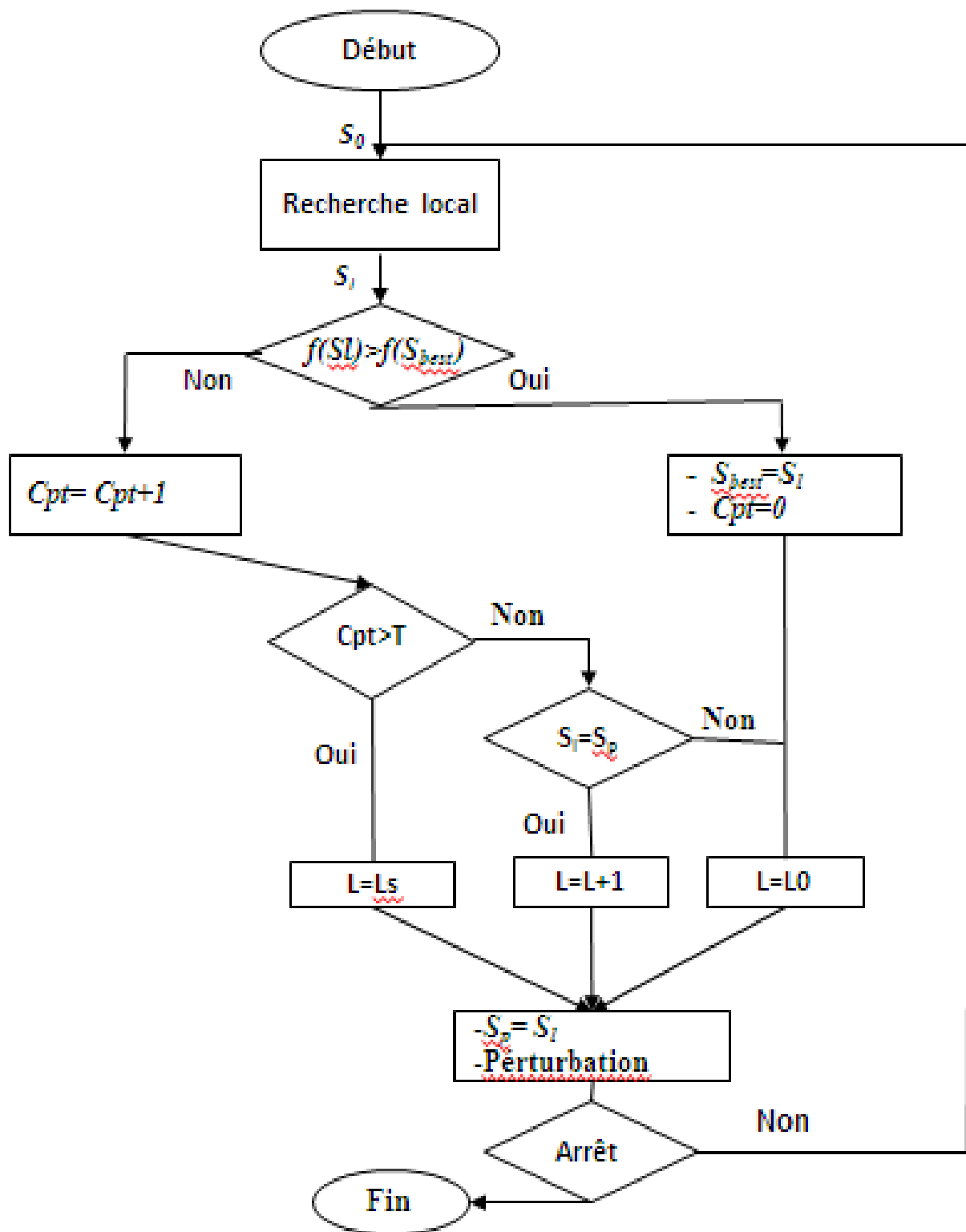


Figure 3.1 : Organigramme du fonctionnement de l'algorithme BLS.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Si la valeur de la variable Cpt est inférieure au paramètre T alors BLS effectue soit une perturbation dirigée avec une probabilité P soit une perturbation non dirigée avec une probabilité $(1-P)$. La formule qui détermine la valeur de P est donnée comme suite :

$$P = \begin{cases} e^{-Cpt/T} & \text{Si } e^{-w/T} > P_0 \\ P_0 & \text{Sinon} \end{cases}$$

Plus la recherche avance sans amélioration plus la valeur P diminue ce qui signifie diminution de la probabilité d'utiliser une perturbation dirigée et augmentation de la probabilité d'utiliser une forte perturbation.

Chaque mouvement effectué (dans la recherche locale ou dans la perturbation) est sauvegardé dans la liste tabou. Par contre le statut tabou d'un mouvement n'est considéré que dans la phase de perturbation, plus précisément dans la perturbation dirigée afin de ne pas permettre une détérioration importante de la qualité de la solution. Le statut tabou d'une solution est ignoré si celle-ci est de meilleure qualité que la meilleure solution (critère d'aspiration).

Le cadre de l'algorithme BLS et ILS est le même en ce qui concerne la manière d'intensification et de diversification qui est la recherche locale et la perturbation. Par contre l'algorithme BLS diffère de l'algorithme ILS lors qu'il emploie une technique de perturbation à plusieurs méthodes, où chaque méthode est appelée suivant l'état de la recherche ce qui mène à plusieurs niveaux de diversification.

BLS diffère aussi de l'algorithme ILS par la non utilisation de la fonction critère d'acceptation lors qu'un optimum local est atteint, en effet, BLS choisit toujours le nouvel optimum local comme nouveau point de départ.

BLS utilise aussi la notion de liste tabou empruntée de la recherche taboue, lors de la perturbation afin de ne pas défaire les mouvements effectués pendant un certain nombre d'itérations et d'empêcher la recherche de revenir à des zones de recherches déjà visitées. Et tout comme la recherche taboue BLS rencontre plusieurs optimaux locaux lors de la recherche, par contre BLS diffère de la recherche taboue dans la diversification qui ne se fait que lors qu'un optimum est atteint. En plus, le critère tabou d'un mouvement n'est vérifié que lors de la phase de perturbation.

BLS (Breakout Local Search)

Entrée : Une instance du problème, condition d'arrêt

Sortie : La meilleure solution trouvée

Paramètres :

L_0 : degré du saut initial.

L_s : degré du saut lors d'une forte perturbation.

T : nombre maximum d'optimums locaux visités sans améliorer la meilleure solution déjà trouvée.

Début

S = GénérerSolutionInitial()

$S_{meilleur} = S$ /* $S_{meilleur}$ stock la meilleur solution trouvée */

$S_p = S$ /* S_p : la solution optimale obtenue lors de la dernière recherche local */

Cpt=0 /* Nombre de fois où la solution optimal n'a pas été améliorée */

Tantque (condition d'arrêt non satisfaite) **faire**

S=RechercheLocal(S)

Si (Evauler(S) >Evauler($S_{meilleur}$)) **Alors**

$S_{meilleur} = S$ /* m-à-j de la meilleur solution */

Cpt=0

Sinon

Cpt = Cpt + 1

FinSi

Si (Cpt > T) **Alors**

/* La recherche est stagnée une forte perturbation est nécessaire */

$L = L_s$

Sinon

Si (S = S_p) **Alors**

/* La recherche est retournée vers le précédant optimum */

```
/* Le degré de perturbation est augmenté*/  
L = L + 1  
Sinon  
L = L0  
FinSi  
FinSi  
  
Sp = S  
S = PerturberSolution( S, L , Cpt )  
Fait ;  
Fin
```

7.1.2.3 Algorithme de perturbation

Algorithme PerturberSolution

Entrée : solution, L : degré du saut, Cpt

Sortie : solution perturbée

Début

Si (Cpt > T) **alors** /* Forte perturbation */

Pour i=0 jusqu'à i= L **faire**

Choisir une BS aléatoirement et changer son état

H ← i + γ // Mettre à jour la liste tabou

Fait ;

Sinon /* Faible perturbation*/

Avec une probabilité égale à P*Q

Pour i=0 jusqu'à i= L **faire**

Faire un mouvement de type P1

H ← i + γ // Mettre à jour la liste tabou. γ est la durée tabou.

Fait

Avec une probabilité égale à P*(1-Q)

<p>Pour $i=0$ jusqu'à $i=L$ faire</p> <p style="padding-left: 40px;">Faire un mouvement de type P2</p> <p style="padding-left: 40px;">$H \leftarrow i + \gamma$ // Mettre à jour la liste tabou</p> <p>Fait</p> <p>Avec une probabilité égale à $1-P$</p> <p>Pour $i=0$ jusqu'à $i=L$ faire</p> <p style="padding-left: 40px;">Choisir une BS aléatoirement et changer son état</p> <p style="padding-left: 40px;">$H \leftarrow i + \gamma$</p> <p>Fait</p> <p>Finsi</p> <p>Fin</p>

8 Notre approche

Notre algorithme comme BLS s'inspire aussi de l'algorithme ILS, lorsqu'il utilise la recherche locale comme moyen d'intensification et un mécanisme de perturbation comme moyen de diversification. Le point de divergence avec ILS consiste dans la manière de choisir le nouveau point de départ après chaque recherche local, il consiste aussi dans l'utilisation d'un mécanisme de réinitialisation de la recherche ainsi dans le moyen de choisir la nouvelle solution du point de départ lors de l'initialisation.

Notre algorithme
<p>Entrée : Une instance du problème, condition d'arrêt</p> <p>Sortie : La meilleure solution trouvée</p> <p>Paramètres :</p> <p>P : nombre de fois sans amélioration avant d'incrémenter la force de perturbation.</p> <p>I : nombre de fois sans amélioration avant la réinitialisation de la recherche.</p> <p>Début</p>

```
S = GénérerSolutionInitial()
SGbest =S /* SGbest stock la meilleur solution trouvée pendant toute la
recherche*/
SLbest =S /* SLbest stock la meilleur solution trouvée depuis la dernière
initialisation*/
Cpt=0 /* Nombre de fois où la solution optimal local n'a pas été améliorée */
Tantque (condition d'arrêt non satisfaite) faire

    S=RechercheLocal(S)
    Si (S est meilleur que SLbest) Alors
        SLbest =S /* m-à-j de la meilleur solution local*/
        Cpt=0
    Sinon
        Cpt = Cpt + 1
    FinSi

    Si (P < Cpt < I) Alors
        /* La recherche est stagnée alors une augmentation de la puissance de
        perturbation est nécessaire */
        Strenght= strenght+1
    FinSi
    S= Perturber(S, strenght)

    Si (SLbest est meilleure que SGbest) Alors
        SGbest=SLbest
    FinSi
    Si ( Cpt > I) Alors
        S= InitialiserSolution(SGbest, nbEval)
        Strength= 1
        Cpt=0
        SLbest =S
    FinSi
Fait ;
Fin
```

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Notre algorithme effectue à chaque itération une recherche locale à partir d'une solution de départ pour en obtenir un optimum local. Ensuite une nouvelle solution de départ pour la prochaine itération est choisie entre cet optimum local et l'optimum local obtenu lors de l'itération qui précède l'itération courante, l'algorithme choisit toujours la meilleure solution des deux. La nouvelle solution est perturbée afin de s'échapper du point d'attraction.

Après un certain nombre d'itération P sans aucune amélioration de la meilleure solution trouvée, la force de perturbation est augmentée en incrémentant le paramètre *strenght*.

Après un certain nombre d'incrémentations de la force de perturbation sans amélioration de la meilleure solution trouvée une réinitialisation de la recherche est effectuée.

Notre algorithme diffère par rapport aux autres algorithmes dans le moment du lancement de l'initialisation, tel que notre algorithme utilise deux variables, une appelée SG_{best} contenant la meilleure solution trouvée pendant toute la recherche et l'autre variable appelée SL_{best} contenant la meilleure solution trouvée depuis la dernière initialisation de la recherche. La réinitialisation de la recherche se fait par rapport à la variable SL_{best} . C.à.d. qu'après un certain nombre d'itération sans amélioration de SL_{best} la recherche est initialisée. Cette manière est utile par rapport à l'initialisation qui se base sur l'amélioration de la meilleure solution trouvée (SG_{best}) lors que la recherche avance est que SG_{best} devienne une solution d'une bonne qualité, là ça devient rare les fois où SG_{best} est améliorée, ce qui conduit à chaque fois à une réinitialisation empêchant l'approfondissement de la recherche et éliminant des pistes qui mènent peut être vers de meilleurs résultats.

La réinitialisation basée sur la meilleure solution trouvée depuis la réinitialisation qui la précède permet de continuer la recherche dès qu'il y'a une amélioration de cette solution.

Algorithme d'initialisation InitialiserSolution

```
Entrée : la meilleure solution trouvée bestSol, nombre d'évaluations effectuées nbEval  
Sortie : nouvelle solution  
Début  
Div = maxEval / 10 /*maxEval est le nombre maximum d'évaluation pour arrêter la recherche*/  
Solution = générer une solution aléatoirement  
Pour chaque élément i de Solution Faire  
    Soit rand un nombre aléatoire entre 1 et 10  
    Si (rand < nbEval/div) alors  
        solution[i]= bestSol[i]  
    FinSi  
Fait  
Fin
```

Notre algorithme d'initialisation permet de générer des solutions aléatoire mais qui en fonction du stade de la recherche s'approche de la meilleure solution trouvée, c.-à-d. si on est au début de la recherche la solution générée est plus aléatoire mais plus la recherche avance, la solution générée s'approche plus de la meilleure solution trouvée jusque-là.

9 Phases de résolution

La résolution du problème APP comporte trois phases: prétraitement, traitement, et post traitement.

9.1 Phase de prétraitement

Dans cette première phase, nous procédons tout d'abord à l'élimination des solutions dont on sait à l'avance qu'elles sont de mauvaise qualité. On se basant seulement sur le nombre d'antennes employé par une solution, on peut savoir des fois si cette dernière est de mauvaise qualité (on ne peut rien savoir si elle est de bonne qualité). En effet, on peut estimer des fois à l'avance le nombre minimum et maximum d'antennes dont toutes solutions qui emploie un nombre d'antennes hors de

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

cet intervalle est de mauvaise qualité. Ainsi on n'a pas besoin d'évaluer ces solutions. Par exemple, si une solution n'emploie qu'une seule antenne, on sait à priori que c'est une mauvaise solution. Et pareil pour les solutions qui utilisent un nombre important d'antennes.

Par conséquent beaucoup de solutions seront éliminées et l'espace de recherche diminué, ce qui permet de diminuer la complexité du problème.

Pour un problème de 1000 emplacements de BS possibles, nous avons un espace de recherche de taille 2^{1000} solutions possibles. Le nombre de solutions éliminées est donné par la formule suivante :

$$S_{\text{éliminées}} = \sum_{k=0}^{k=B_i} C_n^k + \sum_{k=B_s}^{k=n} C_n^k$$

B_i : borne inférieure où toute solution qui a un nombre de BS activées inférieur à ce paramètre est éliminée.

B_s : borne supérieure où toute solution qui a un nombre de BS activées supérieur à ce paramètre est éliminée.

Algorithme prétraitement

Entrée : Une solution S donnée.

Sortie : solution maintenue ou pas

Début

pour $i=0$ **jusqu'à** $i=$ taille de S **faire**

si ($S[i] == 1$) **alors**

$cpt = cpt+1$

FinSi

Fait ;

Si ($25 < cpt < 100$) **alors**

$Acceptée = \text{vrai}$

Sinon

$Acceptée = \text{false}$

FinSi

9.2 Phase de traitement

Dans cette phase, l'espace de recherche ne contient que les solutions qui ont un nombre de BS raisonnable (appartient à l'intervalle $[B_i, B_s]$). L'algorithme de résolution sera appliqué sur ce nouvel espace de recherche.

Le résultat de cette phase est la meilleure solution rendu par l'algorithme appliqué.

9.3 Phase de post-traitement

Dans cette phase nous appliquons une dernière configuration de la solution résultante de la phase précédente. Généralement, la meilleure solution rendue par l'algorithme de recherche se rapproche de la plus meilleure solution (celle dont aucune solution qui peut être trouvée n'est meilleur qu'elle). Ainsi, les BSs activée de la meilleure solution trouvée et ceux de la plus meilleure solution dont on cherche, doivent être proche les uns des autres (où les mêmes). Donc dans cette étape, pour chaque BS activée, on teste si une meilleure solution peut être trouvée si l'on active à sa place une autre BS proche. L'algorithme suivant illustre le fonctionnement de cette phase.

Algorithme de la phase post-traitement
Entrée : Solution rendue par l'algorithme de résolution
Sortie : Solution améliorée
Début
St=S
St[i]=0
Pour chaque BSi activée faire
Pour chaque BSj voisin de BSi faire
St[j]= 1
Si (Evaluer(St)> Evaluer(S)) Alors
S=St
Sinon
St[j]=0

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

<p style="text-align: center;">Fsi</p> <p style="text-align: center;">Fait</p> <p style="text-align: center;">Fait</p> <p style="text-align: center;">Fin</p>

10 Conclusion

Dans ce chapitre nous avons montré comment le problème APP est modélisé, par la méthode des hypergraphes. Ensuite, nous avons décrit les algorithmes qui vont être utilisés pour la résolution, à savoir l'algorithme ILS et l'algorithme BLS ainsi que notre algorithme proposé. Nous avons aussi défini une nouvelle fonction de voisinage permettant de prendre en considération les informations de localisation des stations de base.

Nous verrons dans le chapitre suivant, les résultats rendus par nos méthodes de résolutions appliqués sur différents benchmarks.

Chapitre 4 :

Résultats expérimentaux

1 Introduction

Dans ce chapitre nous présenterons les résultats obtenus par les algorithmes BLS et ILS qui ont été présentés dans le chapitre précédent, ainsi que les résultats de notre algorithme.

Nous emploierons le benchmark proposé par [25] dans la comparaison des résultats fournis par ces algorithmes, et aussi par ceux fournis par ces derniers avec les résultats fournis par d'autres algorithmes exécutés sur le même benchmark dans le même article. Mais dans un premier lieu nous définirons tous d'abord les critères de comparaisons que nous allons utiliser pour évaluer la performance de chaque algorithme.

2 Benchmark

Afin d'évaluer les performances de chaque algorithme, nous exécuterons chacun d'entre eux sur trois instances, deux d'entre eux sont générées aléatoirement et une est réelle. Des résultats d'autres algorithmes exécutés sur l'instance réelle sont aussi fournis dans [25], ce qui va nous permettre de comparer nos résultats avec d'autres algorithmes non traités dans notre étude.

- Instance 1 (instance réel):

L'instance est une représentation du plan géographique de la ville de Malaga qui a une surface égale à 27,2 Km² est qui a été modélisée en une grille de points de taille 450 X 300 points (135000 points), où chaque point représente une surface de 15 X 15 mètres. Pour les sites candidats ou les stations de bases candidates, 1000 sites candidats (1000 sites / 135000 pts) sont réparties sur la grille. Chaque site candidat est défini par sa position (x , y) dans la grille. Des antennes isotropes sont utilisées de rayon de couverture égal à 30 pts. La zone étudiée possède des endroits où il est impossible d'installer des antennes comme la mer et les montagnes, donc qui ne peuvent pas être couverts, ce qui implique que seulement un taux de 95,522% de zone peut être couvert au maximum.

- Instance n°2 (instance aléatoire):

Instance de taille 400x400 contenant 1200 antennes dispersées aléatoirement dans la zone.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

- **Instance n°3 (instance aléatoire):**

Instance de taille 400x300 contenant 800 antennes dispersées aléatoirement dans la zone.

3 Paramètres

L'étape de paramétrage des algorithmes est une étape clés pour l'obtention de bons résultats. Les paramètres de chaque algorithme ont été fixés expérimentalement pour enfin choisir les valeurs dont les algorithmes rendaient les meilleurs résultats.

Les valeurs des paramètres propres à chaque algorithme ont été fixées comme suite :

Algorithme BLS :

- Degré du saut initial $L_0 = 5$
- Degré du saut lors d'une forte perturbation $L_s = 10$
- Nombre maximum d'optimums locaux visités sans amélioration de la meilleure solution avant de lancer une forte perturbation $T=10$

Algorithme ILS :

- Degré du saut initial $L_0 = 2$
- Nombre d'optimum locaux visités sans amélioration avant d'augmenter la force de perturbation $b=3$
- Nombre maximum d'optimums locaux visités sans amélioration de la meilleure solution avant de réinitialiser la recherche $R=6$

Notre algorithme:

- Degré du saut initial $L_0 = 2$
- Nombre d'optimum locaux visités sans amélioration avant d'augmenter la force de perturbation $b=3$
- Nombre maximum d'optimums locaux visités sans amélioration de la meilleure solution avant de réinitialiser la recherche $R=6$

4 Critères de comparaison entre algorithmes

Deux critères sont utilisés dans l'évaluation de la performance d'un algorithme, la valeur de la fonction objectif et le nombre d'évaluation des solutions.

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

4.1 La valeur de la fonction objectif

Ce critère nous permet de savoir qu'elle est l'algorithme qui permet de rendre les solutions qui ont les valeurs de la fonction objectif f les plus élevées, c'est-à-dire l'algorithme qui permet de rendre les solutions de meilleures qualités.

4.2 Nombre d'évaluations

Cette mesure représente le nombre de fois où la fonction d'évaluation d'une solution est appelée pendant toutes l'exécution jusqu'à l'arrivée à la solution final. Elle remplace la mesure temps d'exécution. En effet, la fonction d'évaluation de la solution est la tâche la plus importante pendant l'exécution de l'algorithme et les autres taches sont négligeable par rapport à cette tache en terme de temps de calcule, donc se baser sur cette mesure permet d'éliminer les différences d'environnements matériel et logiciels où les algorithmes s'exécutent, ce qui permet de comparer les algorithmes sous les mêmes bases indépendamment de l'environnement.

5 Résultats

Nous présentons dans cette section les résultats des trois algorithmes appliqués aux instances présentées ci-dessus.

Nous présentons tout d'abord les résultats obtenus lors ce que la fonction de voisinage utilisant des mouvements de type M1 est employée et lors ce que la fonction de voisinage utilisant des mouvements de type M2 est employée. Nous rappelons que les mouvements de type M1 sont des mouvements non dirigés et ceux de type M2 sont des mouvements dirigés (voir la section 5.3).

Les graphes suivants représentent la valeur de la fonction objectif obtenues en fonction du nombre d'évaluation. Les trois algorithmes ont été appliqués sur la première instance.

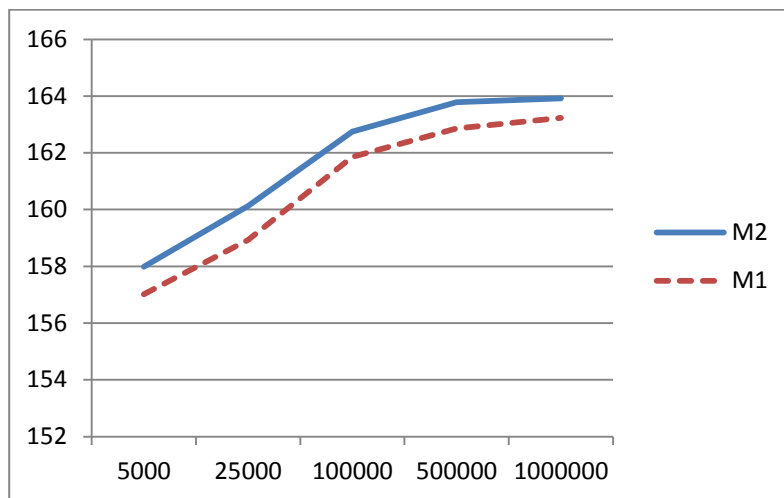


Figure 4.1 : Résultats obtenus sur la première instance par l’algorithme ILS

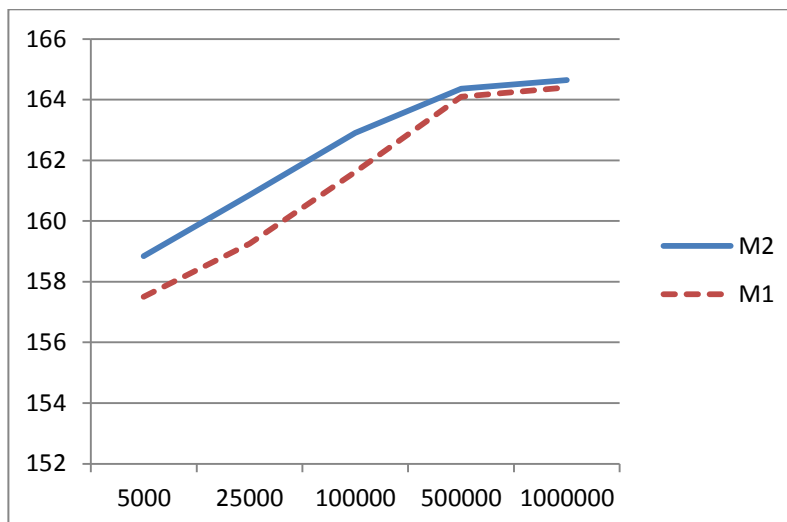


Figure 4.2 : Résultats obtenus sur la première instance par l’algorithme BLS

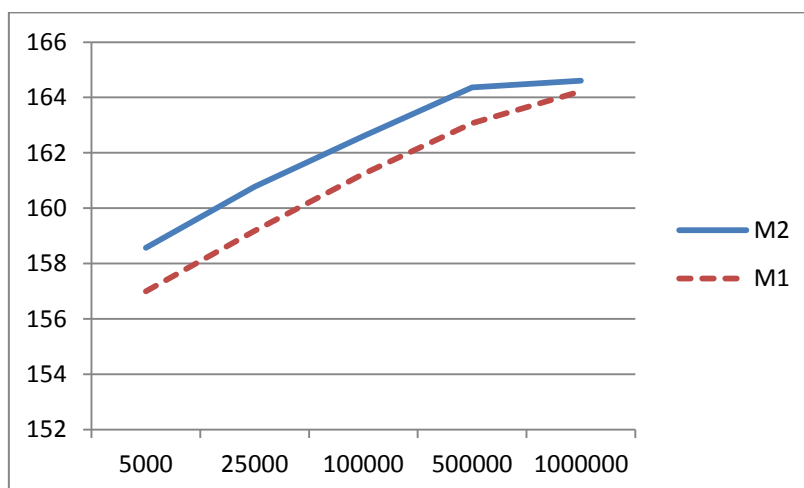


Figure 4.3 : Résultats obtenus sur la première instance par notre algorithme

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

On constate bien que les valeurs de la fonction objectif sont nettement meilleures dans tous les cas lors que la fonction de voisinage qui utilise des mouvements de type M2 (mouvements dirigés) est utilisée. Ces résultats montrent l'importance d'utilisation de mouvements dirigés dans la recherche locale.

Pour la suite de notre travail nous n'avons employé que la fonction de voisinage qui utilise des mouvements de type M2. Les résultats seront présentés sous forme de tableaux où chaque colonne contiendra les valeurs de la fonction objectif pour chaque exécution propre à chaque algorithme. Chaque tableau regroupera les valeurs obtenues par un nombre d'évaluation donné.

5.1 Résultats obtenus sur la première instance

Tableau 1 : Résultats des trois algorithmes pour 25 000 évaluations.

N° Exéc	25 000 Evaluations		
	BLS	ILS	Notre algorithme
1	161.983	161.858	161.115
2	162.230	159.307	159.454
3	162.732	159.201	163.611
4	158.822	159.801	159.795
5	159.047	162.562	160.332
6	160.611	157.402	160.488
7	159.685	157.563	162.187
8	161.263	159.580	160.120
9	159.639	162.978	162.503
10	161.768	161.043	161.139
Moyenne	160,780	160,130	160.861

D'après le tableau 1, les résultats rendus par notre algorithme pour 25000 évaluations dépassent ceux des algorithmes BLS et ILS. La meilleure solution trouvée lors des 10 exécutions et aussi trouvée par notre algorithme lors de l'exécution numéro 3, sa fonction objectif est égale à 163,611. En terme de résultats obtenus lors

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

des 10 exécutions, l'algorithme BLS se classe en deuxième position et l'algorithme ILS en dernière position.

Tableau 2 : Résultats des trois algorithmes pour 100 000 évaluations.

100 000 Evaluations			
N° iter	BLS	ILS	Notre algorithme
1	162.466	163.041	163.076
2	164.047	163.620	163.381
3	164.252	162.441	162.977
4	161.998	160.890	162.176
5	159.187	161.414	162,448
6	162.548	163.112	162,742
7	162.910	163.209	163,286
8	163.377	162.683	163,186
9	161.575	162.295	162,842
10	163.685	164.701	163,025
Moyenne	162,60	162,741	162.913

D'après le tableau 2, les résultats rendus par notre algorithme là aussi dépassent ceux des algorithmes BLS et ILS. La meilleure solution trouvée lors des 10 exécutions quant à elle est trouvée par l'algorithme ILS et est de fonction objectif égale à 164.701. En termes de classement, l'algorithme ILS se classe en deuxième position avant l'algorithme BLS.

Tableau 3 : Résultats des trois algorithmes pour 500 000 évaluations.

500 000 Evaluations			
N° Exéc	BLS	ILS	Notre algorithme
1	164,611	163,521	164.764
2	163,787	164,650	164.045
3	164,779	163,708	163.905
4	164.282	162,786	164.686

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

5	164,279	164,242	164.429
6	164,507	163,973	164,189
7	164,634	164,598	164,665
8	164,629	163,295	163,957
9	163,538	162,693	164,359
10	164,512	164,348	164,665
Moyenne	164,364	163,781	164.366

D'après le tableau 3, les résultats rendus par notre algorithme sont presque égale à ceux rendus par l'algorithme BLS. ILS est en troisième position. La meilleure solution trouvée lors des 10 exécutions quant à elle est trouvée par l'algorithme BLS et est de fonction objectif égale à 164.779.

Tableau 4 : Résultats des trois algorithmes pour 1 000 000 évaluations.

N° iter	1 000 000 Evaluations		
	BLS	ILS	Notre algorithme
1	164.445	163.502	164.701
2	164.692	163.511	164.467
3	164.610	163.888	164.701
4	164.686	163.665	164.701
5	164,508	163.973	164.562
6	164,641	163.620	164.697
7	164,668	163,586	164.682
8	164,679	164,045	164.574
9	164,574	164,258	164.698
10	164,578	163,756	164.639
Moyenne	164,608	163,911	164.642

D'après le tableau 4, les résultats rendus par notre algorithme dépassent légèrement ceux de BLS, et la aussi l'algorithme ILS se classe en dernière position. La meilleure solution obtenue lors des dix exécutions est celle rendue par notre algorithme lors des exécutions 1, 3 et 4.

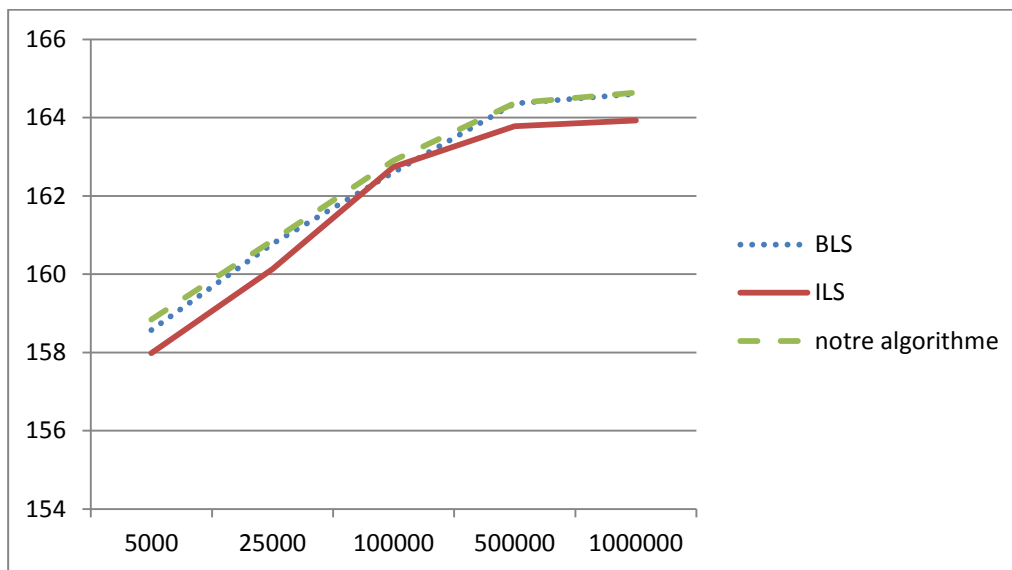


Figure 4.4 : Comparaison entre les trois algorithmes par rapport à la moyenne de la fonction objectif obtenue lors des dix exécutions

On remarque que malgré que la meilleure solution n'est pas toujours rendue par notre algorithme, mais en terme de moyenne notre algorithme dépasse en général les algorithmes ILS et BLS.

Pour le temps d'exécution, nous avons fixé la qualité de solution à obtenir à $f=164.7$. Le nombre moyen d'évaluations nécessaire à chaque algorithme pour arriver à une solution de cette qualité est illustré dans l'histogramme suivant.

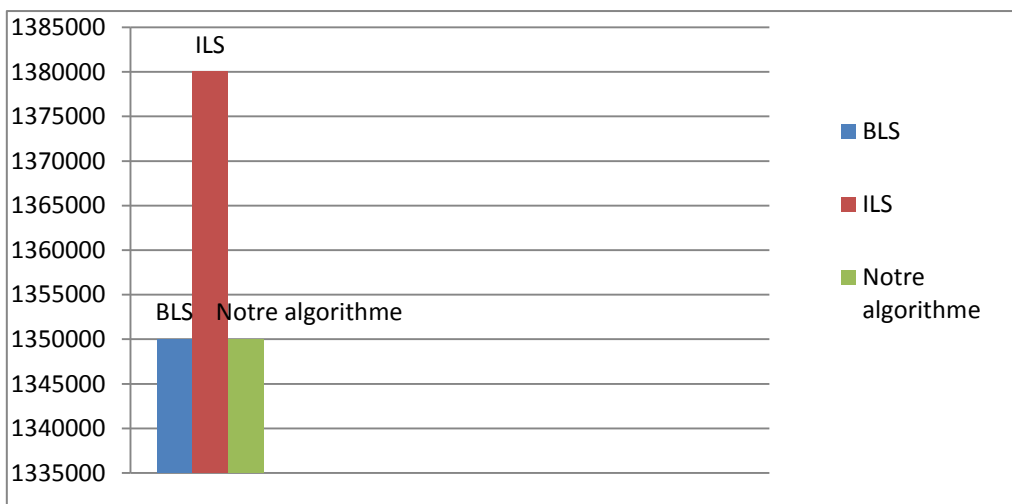


Figure 4.5 : Comparaison du nombre d'évaluations nécessaire pour chaque algorithme pour la meilleure solution

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Notre algorithme et l'algorithme BLS permettent d'offrir une convergence plus rapide que l'algorithme ILS pour cette instance. En effet il ne leurs suffit comme moyenne que 1350000 évaluations pour arriver à la solution qui a été fixée. Par contre l'algorithme ILS lui a fallu 1380000 évaluations.

5.2 Résultats obtenus sur la deuxième instance

Tableau 5 : Résultats des trois algorithmes pour 25 000 évaluations.

25 000 Evaluations			
N° iter	BLS	ILS	Notre algorithme
1	142,883	141,355	142,401
2	141,352	140,805	144,052
3	141,185	143,229	143,990
4	142,066	141,589	142,592
5	141,256	140,984	143,481
6	141,648	142,892	141,871
7	143,058	142,527	143,151
8	144,192	143,246	142,858
9	141,167	140,235	143,256
10	142,293	141,109	143,981
Moyenne	142,110	141,797	143,163

Tableau 6 : Résultats des trois algorithmes pour 125 000 évaluations.

125 000 Evaluations			
N° iter	BLS	ILS	Notre algorithme
1	142,698	144,389	144,408
2	144,887	145,709	142,956
3	143,750	142,889	143,689
4	143,778	143,332	144,765
5	142,978	143,926	143,444
6	144,159	144,018	142,889

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

7	143,869	144,043	143,691
8	144,268	145,123	144,269
9	142,756	143,168	142,998
10	144,578	143,843	143,812
Moyenne	143.772	144.044	143.692

Tableau 7 : Résultats des trois algorithmes pour 525 000 évaluations.

N° iter	525 000 Evaluations		
	BLS	ILS	Notre algorithme
1	145,908	145,663	143,704
2	145,425	145,04	145,996
3	144,929	144,763	145,495
4	145,268	145,648	145,068
5	144,869	144,869	145,321
6	145,587	145,256	145,236
7	145,635	145,259	144,489
8	144,948	144,769	145,176
9	145,763	145,263	145,598
10	145,861	144,984	145,621
Moyenne	145.419	145.151	145.170

Tableau 8 : Résultats des trois algorithmes pour 1 025 000 évaluations.

N° iter	1 025 000 Evaluations		
	BLS	ILS	Notre algorithme
1	145,838	145,448	146,078
2	145,331	145,73	145,331
3	145,584	145,589	145,623
4	145,395	145,635	146,041
5	145,489	145,626	145,763
6	145,729	145,427	145,723
7	145,692	145,589	145,896

8	145,623	145,703	145,704
9	145,584	145,472	145,409
10	145,577	145,643	145,389
Moyenne	145,5842	145,5862	145,6957

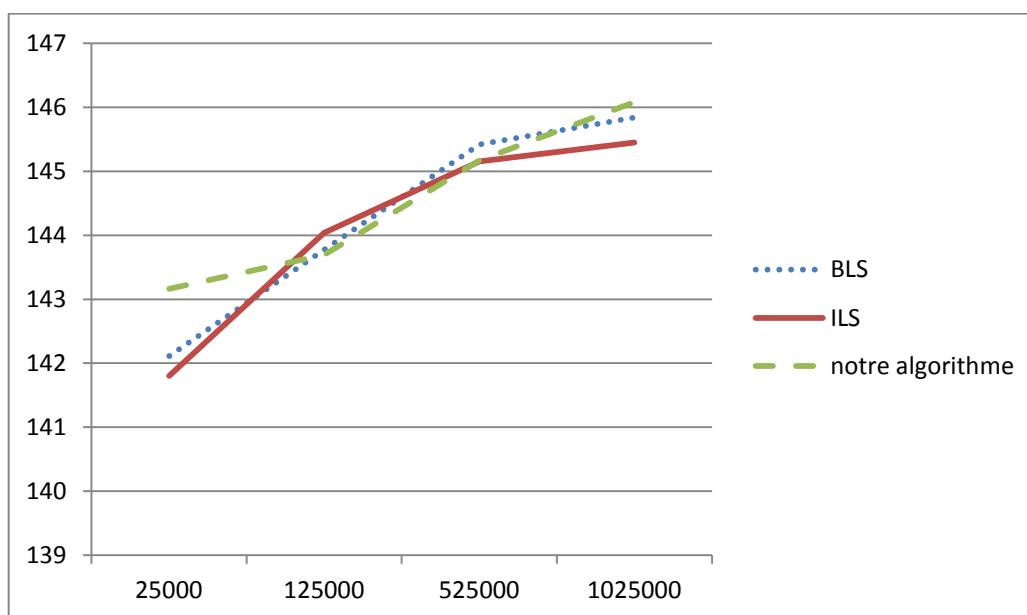


Figure 4.6 : Comparaison entre les trois algorithmes par rapport à la moyenne de la fonction objectif obtenue lors des dix exécutions

On remarque d'après le graphe 4.3 que les résultats des trois algorithmes se rapprochent presque pour tous nombres d'évaluations sauf pour les expérimentations à 25000 évaluations où notre algorithme présente un nette avantage.

En ce qui concerne le temps d'exécution, nous avons fixé la qualité de solution à obtenir à $f= 146$. Le nombre moyen d'évaluations nécessaire à chaque algorithme pour arriver à une solution de cette qualité est illustré dans l'histogramme suivant.

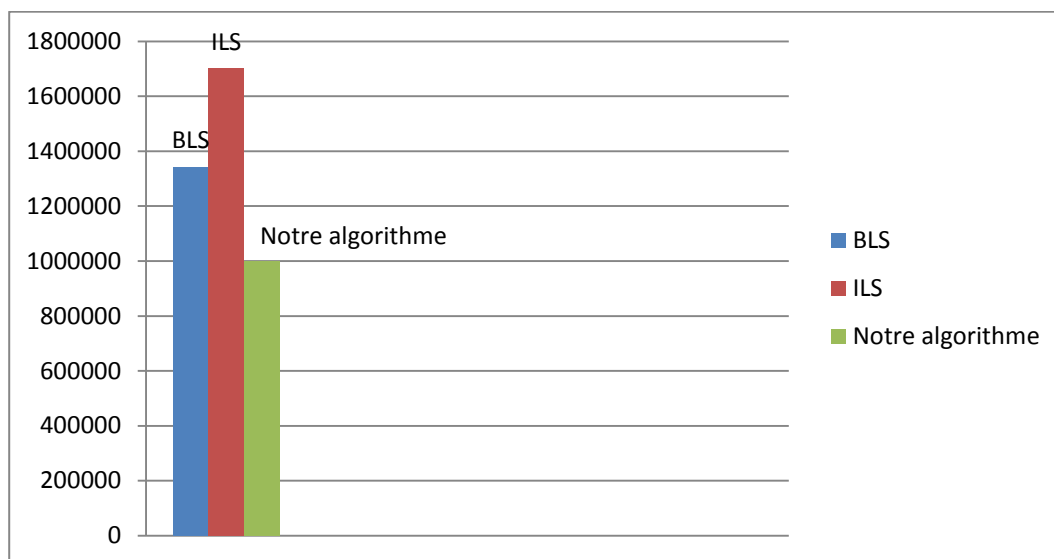


Figure 4.7 : Comparaison du nombre d'évaluations nécessaire pour chaque algorithme pour atteindre la meilleure solution

Là aussi notre algorithme a été le plus rapide par rapport aux deux autres algorithmes avec un nombre moyen d'évaluations égale à 1 000 000 l'algorithme BLS est en deuxième place avec une moyenne de 1 340 000 évaluations, par contre ILS nécessite 1 700 000 évaluations.

5.3 Résultats sur la troisième instance

Tableau 9 : Résultats des trois algorithmes pour 25 000 évaluations.

N° iter	25 000 Evaluations		
	BLS	ILS	Notre algorithme
1	181,216	182,165	179,083
2	183,709	182,893	183,278
3	182,842	183,193	183,101
4	182,594	182,945	181,452
5	182,467	183,052	182,587
6	183,189	182,683	183,126
7	181,861	182,194	183,122
8	183,121	183,128	182,853
9	182,169	182,204	181,023
10	182,689	183,051	182,263

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

Moyenne	182,5857	182,7508	182,1888
---------	----------	----------	----------

Tableau 10 : Résultats des trois algorithmes pour 125 000 évaluations.

125 000 Evaluations			
N° iter	BLS	ILS	Notre algorithme
1	185,341	185,215	186,915
2	185,264	185,473	185,481
3	182,764	183,317	186,817
4	184,868	184,668	186,425
5	183,685	183,925	185,588
6	184,698	184,842	185,904
7	185,052	185,175	185,542
8	184,263	184,891	185,621
9	183,457	183,872	186,204
10	185,122	185,268	184,804
Moyenne	184,4514	184,6646	185,931

Tableau 11 : Résultats des trois algorithmes pour 525 000 évaluations.

525 000 Evaluations			
N° iter	BLS	ILS	Notre algorithme
1	187,026	185,828	185,814
2	186,044	185,814	186,784
3	185,526	187,611	185,006
4	186,658	186,578	185,965
5	186,266	186,269	186,896
6	187,002	187,294	187,212
7	185,631	185,896	186,468
8	186,065	186,417	186,806
9	185,832	187,517	187,234
10	185,898	185,717	186,471

Moyenne	186,1948	186,4941	186,4656
---------	----------	----------	----------

Tableau 12 : Résultats des trois algorithmes pour 1 025 000 évaluations.

1 025 000 Evaluations			
N° iter	BLS	ILS	Notre algorithme
1	186,256	187,635	186,818
2	186,371	185,799	187,635
3	186,692	186,869	186,925
4	185,857	186,289	186,885
5	186,312	186,265	186,963
6	186,256	185,965	187,541
7	186,896	187,521	187,628
8	187,134	186,982	186,723
9	185,896	186,614	187,583
10	186,128	187,256	187,302
Moyenne	186,3798	186,7195	187,2003

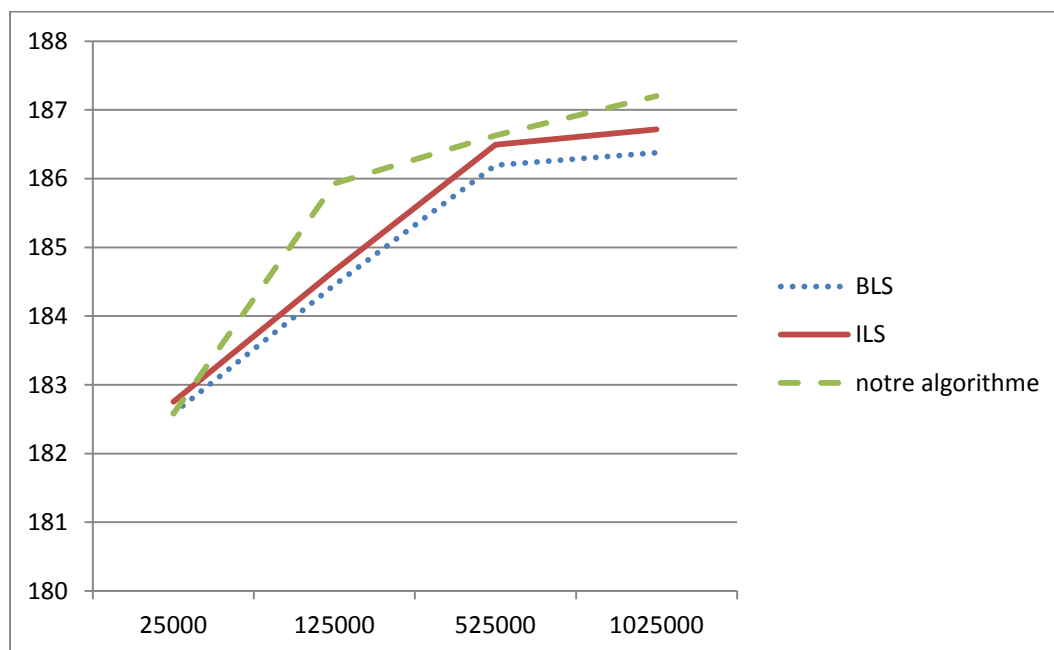


Figure 4.8: Comparaison entre les trois algorithmes par rapport à la moyenne de la fonction objectif obtenue lors des dix exécutions

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

On remarque d'après le graphe 4.5 que notre algorithme présente des résultats nettement meilleurs que ceux des deux autres algorithmes, même si ceux rendus par 25 000 évaluations sont presque identiques. L'algorithme ILS est en deuxième place pour cette fois.

En fixant la qualité de solution à $f= 187.2$, le nombre moyenne d'évaluations nécessaire à chaque algorithme pour arriver à une solution de cette qualité est illustré dans l'histogramme suivant.

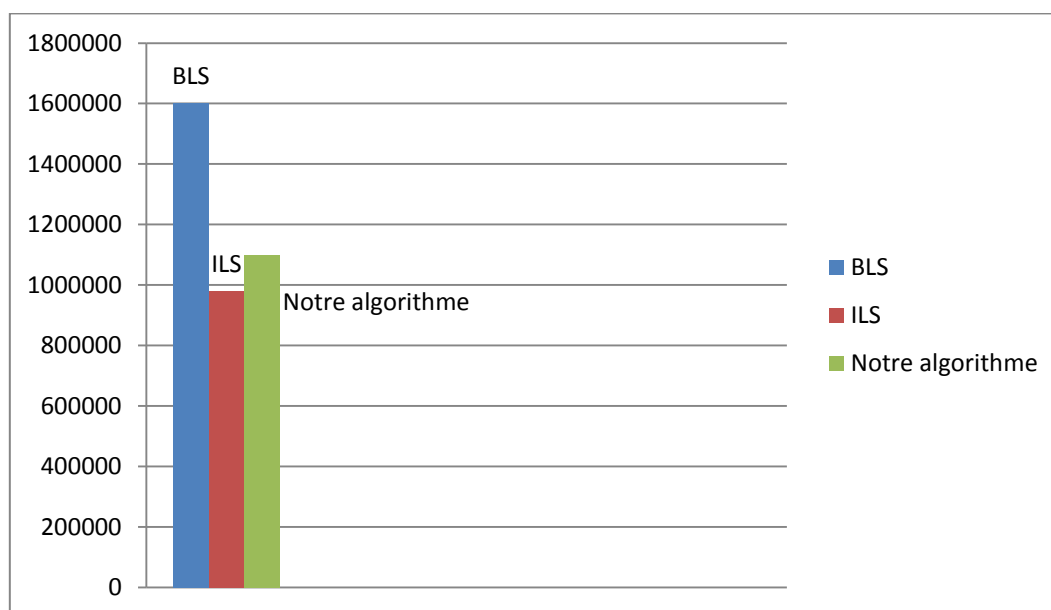


Figure 4.9 : Comparaison du nombre d'évaluations nécessaire pour chaque algorithme pour la meilleure solution

Pour cette dernière instance, c'est l'algorithme ILS qui performe le meilleur résultat en ce qui concerne le temps d'exécution avec un nombre moyen d'évaluation égale à 980 000. Notre algorithme est en deuxième place avec une moyenne égale à 1100 000. L'algorithme BLS nécessite 1600 000 évaluations en moyenne.

Par rapport aux résultats obtenus sur les trois instances on peut dire que notre approche présentée des résultats très satisfaisants en termes de qualité de solutions obtenues et de temps d'exécution. Où ces résultats ont dépassés ceux des algorithmes ILS et BLS un bon nombre de fois.

6 Comparaison avec les résultats obtenus dans le travail de Mendes et al

La figure ci-dessous présente les valeurs de la fonction objectif obtenues par plusieurs algorithmes testés dans le travail de Mendes et al [25] sur leurs benchmark décrit dans la section 2 de ce chapitre. On lui a ajouté les valeurs de la fonction objectif des trois algorithmes implémentés dans ce travail.

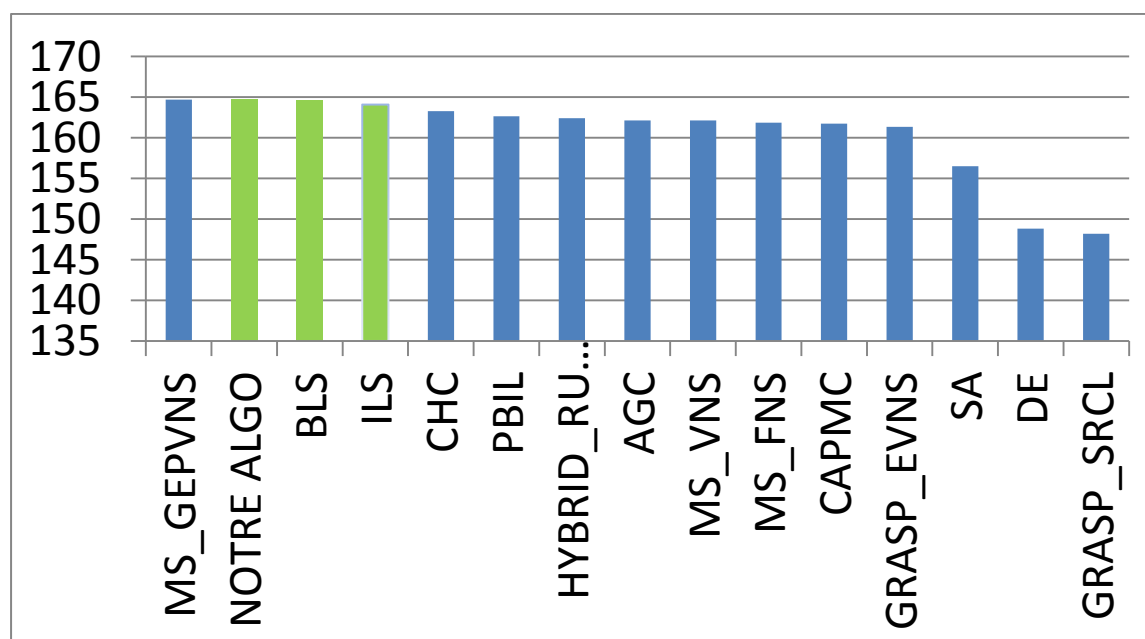


Figure 4.10 : Résultats des différents algorithmes testés dans [25] ainsi que nos trois algorithmes (en vers)

Appliqués sur la même instance, nos trois algorithmes sont classés respectivement deuxième troisième et quatrième. MS_GEPVNS qui est une version améliorée de l'algorithme VNS est le seul algorithme qui dépasse nos trois algorithmes en termes de qualité de solution.

Notre algorithme est classé en deuxième position et dépasse des algorithmes comme le recuit simulé SA, l'AGC et le CHC.

7 Conclusion

Nous avons présenté dans ce chapitre les résultats obtenus par les trois algorithmes présentés lors du chapitre précédent. Les algorithmes ont été appliqués sur une instance réelle offerte par [25]. Nous avons montré que notre algorithme fournit de très bons résultats. Les algorithmes ont été comparés par rapport à la valeur de la fonction objectif des solutions rendues. Nous avons défini une fonction de voisinage dirigée par des informations liées au problème traité et nous avons montré que les solutions obtenues par cette dernière sont nettement meilleures que ceux obtenues lors qu'une fonction de voisinage non dirigée est employée. Nous avons utilisé comme critère d'arrêt le nombre maximum d'évaluation à effectuer et nous avons montré aussi que ce critère remplace le critère temps d'exécution et faisait abstraction à toutes contraintes techniques indépendantes du problème.

Conclusion générale

Les métaheuristiques figurent parmi les méthodes les plus adaptées dans la résolution des problèmes d'une complexité importante. Les métaheuristiques de trajectoires ou de voisinage sont des méthodes qui ont prouvées leur efficacité lors que appliquées à divers problèmes inclus aussi le problème étudié dans notre travail c.à.d. le problème de positionnement d'antennes en réseaux GSM (APP). Ces méthodes même si elles nécessitent moins de temps d'exécution que plusieurs autres méthodes (méthodes évolutionnaires par exemple) elles rendent des résultats remarquables. C'est ainsi que nous nous sommes intéressés à ces méthodes qui sont basés sur le principe de la recherche local.

Nous nous sommes intéressé dans notre travaille aux algorithmes ILS et BLS qui s'inspire du premier. Ces deux algorithmes qui ont donné de très bons résultats auparavant. Nous avons aussi proposé un nouvel algorithme qui s'inspire aussi de l'algorithme ILS. Nous avons aussi montré la nécessité d'utiliser des fonctions de voisinage dirigées par des informations liés au problème, ainsi nous avons présenté notre fonction qui permet le passage d'une solution à son voisin. Nous avons montré à travers les expérimentations l'importance de l'utilisation d'un mécanisme de perturbation adaptable comme celui utilisé par BLS. Nous avons montré aussi les bonnes performances de notre algorithme.

Nous envisageons comme perspectives le teste de notre algorithme sur d'autres instances notamment des instances plus complexe et plus réelle en utilisant des modèles de propagation d'ondes d'antennes réelles, compléter le travaille par des approches multiobjectif et essayer d'utiliser le concept d'hybridation de méthodes

RÉFÉRENCES BIBLIOGRAPHIQUES

Bibliographie

- [1] Akella, M.R., Batta, R., Delmelle, E.M., Rogerson, P.A., Blatt, A., Wilson, G.: Base station location and channel allocation in a cellular network with emergency coverage requirements. *European Journal of Operational Research* 164(2), 301–323 (2005)
- [2] Alba, E., 2004. Evolutionary Algorithms for Optimal Placement of Antennae in Radio Network Design. In: IPDPS-NIDISC'04, p. 168. (PPSN VIII), vol. 3242 of Lecture Notes in Computer Science, Springer-Verlag, Birmingham, UK, p. 832-842, 2004.
- [3] E. Alba, G. Molina, and F. Chicano. Optimal placement of antennae using metaheuristics. In *Numerical Methods and Applications (NM&A-2006)*, Borovets, Bulgaria, August 2006.
- [4] Alba, E. and Chicano, F., 2005. On the Behavior of Parallel Genetic Algorithms for Optimal Placement of Antennae in Telecommunications. *International Journal of Foundations of Computer Science*, 16 (2), 343 – 359.
- [5] E. ALBA, J.F. CHICANO, Evolutionary Algorithms in Telecommunications, <http://neo.lcc.uma.es/staff/francis/pdf/melecon06.pdf>, Accessed: 23.11.2011.
- [6] Antonio J. Nebro, Enrique Alb, Guillermo Molina, Francisco Chicano, Francisco Luna, Juan J. Durillo “Optimal Antenna Placement. Using a New Multi-Objective CHC”, 2007.
- [7] J. Bendisch, R. Höns, H. Mühlenbein, and J. Zimmermann. Algorithms for radio network optimisation. In *ARNO-Workshop at EPMCC'99*, Paris, March 1999. GMD
- [8] Una Benlic and Jin-Kao Hao: Breakout Local Search for the Max-Cut Problem, 2012. Accepted to *Engineering Applications of Artificial Intelligence*, Sept. 3, 2012
- [9] U. Benlic, J.K. Hao. A study of breakout local search for the minimum sum coloring problem. To appear in Bui LT et al. (Eds.) *SEAL 2012*, Lecture Notes in Computer Science, 2012.
- [10] P. Calegari, F. Guidec, P. Kuonen, and D. Kobler, “Parallel islandbased genetic algorithm for radio network design,” *J. Parallel Distrib. Comput.*, vol. 47, no. 1, pp. 86–90, 1997.

[11] F. Chiang and R. Braun, "Toward a management paradigm with a constrained benchmark for autonomic communications," in *Lecture Notes Computer in Science*, vol. 4456, New York: Springer-Verlag, 2007, pp. 250–258.

[12] K. Deb. Evolutionary algorithms for multicriterion optimization in engineering design. In *Proc. of Evolutionary Algorithms in Engineering and Computer Science*, EUROGEN'99, 1999.

[13]. Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.

[14] L.J. Eshelman, The CHC adaptive search algorithm: how to safe search when engaging in non traditional genetic recombination, In *Foundations of Genetic Algorithms*, G.J.E. Rawlins (Ed.), Morgan Kaufmann, SanMateo, pp.265-283, 1991.

[15] C.M. Fonseca and P.J. Fleming. Multiobjective genetic algorithms made easy: selection sharing and mating restrictions. In *IEEE Int.Conf. on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 45–52, Sheffield, UK, 1995.

[16] El-Ghazali Talbi, Hervé Meunier, "Hierarchical parallel approach for GSM mobile network design", In *J.Parallel Distrib. Comput.* 66 (2006) 274 – 290

[17] Glover, F. Future Paths for integer Programming and Links to Artificial Intelligence. *Comput. Oper. Res.* 13(5) : 533-549, May 1986.

[18] D.E. Goldberg. Genetic algorithms in search, optimization and machine learning. Addison Wesley, 1989.

[19] B.W. KERNIGHAN, S. LIN, An efficient heuristic for partitioning graphs. *Bell System Technology Journal* 49(2) : 291-307, 1970.

[20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Sci*, vol. 220, no. 4598, pp. 671–680, 1983.

[21] I. Kostanic and N. Faour, "Automatic radio planning of GSM cellular networks," *Nonlinear Anal.*, vol. 63, no. 5–7, pp. e847–e858, 2005.

[22] S. LIN, B.W. KERNIGHAN, An efficient heuristic for the traveling-salesman problem. *Operations Research* 21 : 498-516, 1973.

[23] D. Levine. Users guide to the PGAPack parallel genetic algorithm library. Technical Report ANL-95/18, Argonne National Laboratory, Mathematics and Computer Science Division, January 31 1995.

[24] H. R. Lourenco, O. C. Martin, and T. Stützle, “Iterated local search,” in *Handbook of Metaheuristics*, Boston, MA: Kluwer, 2002, pp. 321–353.

[25] Mendes, S.P., Molina, G., Vega-Rodriguez, M.A., Gomez-Pulido, J.A., Sez, Y., Miranda, G., Segura, C., Alba, E., Isasi, P., Len, C., Snchez-Prez, J.M.: Benchmarking a Wide Spectrum of Meta-Heuristic Techniques for the Radio Network Design Problem. *IEEE Transactions on Evolutionary Computation*, 1133–1150 (2009)

[26] Herv Meunier, El-Ghazali Talbi, and Philippe Reininger. A multiobjective genetic algorithm for radio network optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, pages 317–324, La Jolla Marriott Hotel La Jolla, California, USA, 6-9 2000. IEEE Press.

[27] I.H. Osman and G. Laporte, *Metaheuristics: a bibliography*. *Annals of Operations Research* 63, 513-623, 1996]

[28] Oscar Cord´on, Sergio Damas, and Jos´e Santamar´ia. A chc evolutionary algorithm for 3d image registration. In *Fuzzy Sets and Systems IFSA 2003*, volume 2715/2003 of *Lecture Notes in Computer Science*.

[29] Reininger, P. (1997). “ARNO Radio Network Optimisation Problem Modelling,” ARNO Deliverable N1-A1-Part1. FT. CNET, July 15, 1997.

[30] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, London, 1986.

[31] K. STERGLITZ, P. WEINER, Some improved algorithms for computer solution of the travelling salesman problem. *Proc. of 6th Allerton Conf. on Circuit and System Theory*, Urbana, Illinois, p. 814-921, 1968.

[32] TANESE, R. Distributed Genetic Algorithms. In *Second International Conference on Genetic Algorithms* (1989), J. Schaffer and P. Laboratories, Eds., Morgan Kaufmann.

[33] E. Zitzler and L. Thiele. Tournament selection, niching, and the preservation of diversity. Technical Report 43, Computer Engineering and Networks Laboratory, ETH, Zurich, Switzerland, 1998.

[34]. E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. Giannakoglou,

Une approche à voisinage pour le Problème de Positionnement des Antennes en réseaux GSM.

D. Tsahalidis, J. Periaux, K. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design, optimisation, and Control*, pages 19–26, Barcelona, Spain, 2002. CIMNE.

[35] Zitzler E., Künzli S., « Indicator-Based Selection in Multiobjective Search », *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature*

[36] Carlos Segura, Yanira González, Gara Miranda, and Coromoto León “A Multi-Objective Evolutionary Approach for the Antenna Positioning Problem” 2010.

[37] MICHEL VASQUEZ , JIN-KAO HAO A Heuristic Approach for Antenna Positioning in Cellular Networks. *Journal of Heuristics*, 7: 443–472, 2001 Kluwer Academic Publishers