

Table des matières

Introduction Générale	1
CHAPITRE I :	3
INTRODUCTION A LA RADIO COGNITIVE	3
I. CHAPITRE I : Introduction à la radio cognitive	4
I.1 Introduction.....	4
I.2 Radio Cognitive	4
I.2.1 Historique	4
I.2.2 Définition et principe	5
I.3 RADIO LOGICIEL	7
I.4 L'auto-configuration de la radio cognitive	8
I.4.1 Fréquence d'opération.....	8
I.4.2 Modulation	8
I.4.3 Puissance de transmission	8
I.5 Architecture de la radio cognitive	8
I.6 Architecture des réseaux radio cognitive	9
I.6.1 Le réseau primaire.....	9
I.6.2 Le réseau secondaire	9
I.7 Cycle de cognition	11
I.8 Fonctions RC	12
I.8.1 Gestion spectrale	12
I.8.2 L'analyse spectrale	13
I.8.3 La décision spectrale	13
I.8.4 La mobilité du spectre	13
I.9 Conclusion	14
Chapitre II : Optimisation multi-objectifs, Décision Multicritères.....	15
II. CHAPITRE II : Optimisation multi-objectifs, Décision Multicritères (topsis)	16
II.1 Introduction.....	16
II.2 Dominance de Pareto	16

II.2.1	Définition	16
II.2.2	Problème d'optimisation combinatoire multi objectif	16
II.2.3	Concepts de bases.....	17
II.2.4	Points particuliers.....	17
II.3	Méthode de décision Multicritères : TOPSIS (Technique for Order by Similarity to Ideal Solution).....	20
II.3.1	Matrice de décision	20
II.3.2	Algorithme topsis	21
II.4	Conclusion	22
Chapitre III : méthodes de résolution exacte et Approchées		23
III.	Chapitre III: Méthode de résolution exacte et approchées	24
III.1	Introduction	24
III.2	Les méthodes de résolution exactes	25
III.2.1	Programmation dynamique	25
III.2.2	Programmation dynamique parallèle	26
III.2.3	La méthode Branch and Bound.....	26
III.3	Méthode de résolution Approchées	28
III.3.1	Métaheuristiques	29
III.4	Conclusion	33
Chapitre IV: Implémentation et expérimentation		35
IV.	Chapitre IV: Implémentation et expérimentation.....	36
IV.1	Introduction	36
IV.2	Fonction objectif	36
IV.3	Programmation dynamique	42
IV.4	Algorithme Branch and Bound	49
IV.5	Algorithme génétique	38
IV.5.1	Algorithme général	39
IV.6	Comparaison des approches	53
IV.7	Conclusion	54

Liste des figures

Figure I.1 : Exemple d'utilisation du spectre radio.

Figure I.2 : Exemple des trous spectraux.

Figure I.3 : relation entre radio cognitive et radio logiciel.

Figure I.4 : Coexistence entre deux types réseau : Primaire et secondaire.

Figure I.5 : Le cycle de cognition de Mitola et Maguire.

Figure II.1 : Front Pareto : ensemble solution non dominée.

Figure II.2 : Points particuliers : point Idéal, point nadir.

Figure III.1 : Classification des méthodes d'optimisation.

Figure III.2 : Principe algorithme séparation et évaluation (B&B).

Figure III.3 : Organigramme Algorithme génétique.

Figure III.4 : exemple de croisement à point multiple.

Figure III.5 : exemple de mutation à changement minimum.

Figure IV.1 : résultat en termes temps exécution (ms)

Figure IV.2 : résultat en termes de gain (prix, qualité) (génétique)

Figure IV.3 : principe des threads pour prg dynamique

Figure IV.4 : résultats en termes de gain (prix, qualité) dynamique

Figure IV.5 : Temps exécution par rapport aux nombre de threads

Figure IV.6 : Comparaison temps exécution (dynamique, génétique)

Figure IV.7 : résultats en termes de gain (prix, qualité) B&B

Figure IV.8: comparaison en termes de gain (prix, qualité) (Dynamique, B&B, Génétique)

Liste des tableaux

Tableau IV.1 : tableau récapitulatif des résultats (génétique)

Tableau IV.2 : résultats obtenus triés avec TOPSIS

Tableau IV.3 : temps exécution programmation dynamique parallèle

Tableau IV.4 : temps exécution B&B parallèle

Tableau IV.5 : comparaison temps exécution

GLOSSAIRE

AG	Algorithmes Génétiques
B&B	branch and bound
BF	Bande de fréquence
BS	Bande spectralBande spectral
RC	Radio cognitive
RRC	Réseau de radio cognitive
SDR	Software Defined Radio
TOPSIS	Technique for Order by Similarity to Ideal Solution
PU	utilisateur primaire
SU	utilisateur secondaire
URC	utilisateur radio cognitive
OC	optimisation combinatoire
POC	Problème optimisation combinatoire

Introduction Générale

L'industrie des technologies de l'information et de la communication est aujourd'hui confrontée à un défi mondial. Développer de nouveaux services avec une meilleure qualité de service et en même temps réduire son impact environnemental. Clairement, il y a un besoin profond d'efficacité globale non seulement dans le domaine de l'énergie, mais aussi dans le domaine spectral.

Cependant, l'écart entre l'offre sans fil et les demandes s'élargit. Il est à craindre qu'une crise de spectre imminente, dans laquelle la demande explosive des téléphones intelligents va bientôt submerger la capacité sans fil, se produira. Le problème est le manque de nouveau spectre disponible pour les transporteurs de données sans fil. Le trafic de données des smartphones augmente si rapidement que si rien n'est fait, ils utiliseront le spectre disponible. Cela n'affecte pas seulement les téléphones intelligents, mais aussi tous les appareils sans fil. Une énorme quantité de spectre est nécessaire pour l'utilisation de la large bande à l'avenir, comme le suggèrent le National Broadband Plan (NBP) aux États-Unis et l'agenda numérique en Europe.

Afin de résoudre le problème de l'efficacité d'utilisation du spectre, le concept de radio cognitive (RC) a été proposé. La technologie radio cognitive a le potentiel d'être une force perturbatrice dans la gestion du spectre. Le système radio opportuniste devrait être capable d'exploiter les trous de spectre en les détectant et en les utilisant de manière opportuniste.

Les approches exactes sont très vite limitées, et des métaheuristiques dédiées à la résolution de ces problèmes ont été mises au point, mais la difficulté de ce type de problèmes qui appartient à la classe des problèmes NP-difficile encourage l'utilisation de méthodes d'optimisation parallélisée. Pour cela, nous avons choisi la programmation dynamique et l'algorithme de séparation et évaluation (B&B) comme méthodes exactes, et une métaheuristique qui est l'algorithme génétique comme méthodes approchées, afin d'établir une étude comparative selon différents critères, tels que le temps de traitement, le gain obtenu.

Dans ce PFE, nous tenterons d'apporter une meilleure solution à la gestion du spectre, en termes de précision et temps d'exécution. Pour cela, nous avons opté à des méthodes exactes pour l'optimisation multicritère de la gestion de spectre dans les réseaux de radio cognitive.

Au début Nous avons implémenté deux méthodes exactes la programmation dynamique et la méthode branch and bound, pour la meilleure sélection du spectre ; et puis nous avons réalisé une implémentation parallèle sur ces méthodes exactes pour améliorer le temps d'exécution.

Les résultats ont été comparés avec une métaheuristique, qui est l'algorithme génétique.

Ce mémoire est organisé comme suit :

Le premier chapitre : nous donnerons une présentation claire et détaillée de la RC ainsi que ses composantes et ses fonctionnalités et aussi son architecture et son auto configuration.

Le deuxième chapitre : nous expliquons les concepts de bases et les principales définitions de la dominance de Pareto et la méthode topsis utilisés dans l'optimisation multicritère.

Le troisième chapitre : Nous présentons les méthodes exactes et les méthodes approchées utilisées dans ce rapport, en se focalisant sur leurs fonctionnements.

Le dernier chapitre : nous présentons nos expérimentations avec une étude comparative avec une discussion des résultats obtenus.

CHAPITRE I :
INTRODUCTION A LA RADIO
COGNITIVE

I. CHAPITRE I : Introduction à la radio cognitive

I.1 Introduction

Dans les réseaux sans fil actuels, l'allocation du spectre se fait de manière statique et ce sont des agences gouvernementales qui les allouent aux opérateurs détenteurs de licence. Récemment, dû à l'augmentation du volume d'utilisation du spectre électromagnétique, cette politique d'allocation a conduit à un encombrement dans certaines parties du spectre, tandis qu'une large portion allouée de ces BS sont utilisée sporadiquement, ce qui conduit à une sous-utilisation spectrale [7]. C'est pourquoi des techniques d'accès dynamique au spectre ont été proposées pour résoudre ces problèmes d'inefficacité spectrale. La technologie la plus appropriée pour relever ce défi est la radio cognitive. Cette radio est capable de partager la BS d'une manière opportuniste avec les PU détenteurs de licence et les SU. [4]

Les R.R.C ont la capacité de fournir une qualité plus grande pour les utilisateurs mobiles grâce aux techniques d'accès dynamique et grâce aux architectures des réseaux sans fil hétérogènes. La réalisation de cet objectif (améliorer l'utilisation spectrale) est faisable uniquement par le biais de techniques dynamiques et efficaces de gestion des BS [3].

I.2 Radio Cognitive

Historique

En 1998, à l'institut royal de technologie KTH, Joseph Mitola III expose ses travaux sur le radio qui est consciente de l'environnement électromagnétique, qui est capable de modifier le comportement de sa couche physique et qui peut adopter des stratégies complexes [1] La radio cognitive a fait l'objet de l'intérêt particulier Comme étant une technologie qui peut améliorer les performances des réseaux sans fil et l'efficacité de l'usage des bandes spectral [5].

En 2002, le rapport du groupe de travail sur la politique du spectre de la FCC a identifié que la plupart des fréquences étaient inutilisées la plupart du temps, comme le montre la **figure I.1**.

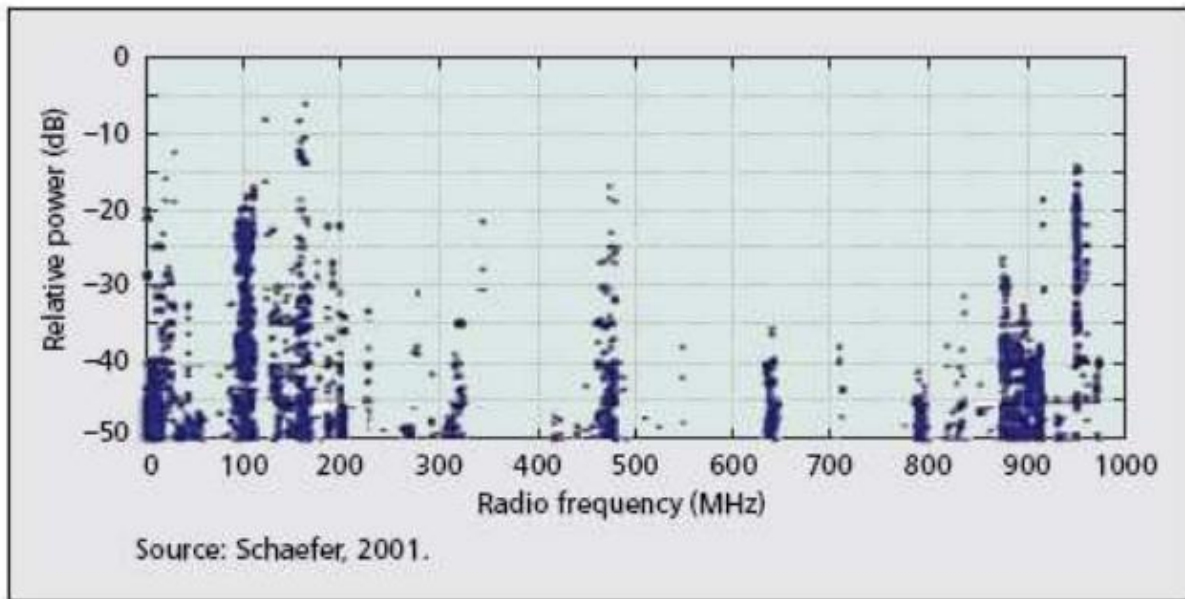


Figure I.1 : Exemple d'utilisation du spectre radio [5]

Diverses mesures de l'utilisation du spectre ont montré que le spectre est sous-utilisé, en ce sens que le cycle de fonctionnement typique de l'utilisation du spectre à une fréquence fixe et à un emplacement géographique aléatoire est faible. Cela signifie qu'il y a de nombreux "trous" dans le spectre radio qui pourraient être exploités.

Ils sont alors rendu compte que la rareté du spectre est principalement déterminée par des systèmes anciens d'attribution de spectre et non par un manque fondamental de spectre.

Définition et principe

Les systèmes de communication sans fil traditionnels ont des paramètres de transmission fixes. En d'autres termes, leur fréquence d'émission est fixe et la même dans tous les lieux et instant de temps, déterminée par des normes réglementaires. La radio cognitive est une technique qui fournit la capacité pour l'utilisateur sans licence ou (SU) utiliser ou partager le spectre d'une manière opportuniste à partir de l'auto-coexistence avec l'utilisateur (PU), en changeant les paramètres de transmission permettant de fonctionner dans le meilleur canal disponible en fonction de comportement de l'environnement. La RC doit déterminer quelle partie du spectre n'a pas d'utilisateurs autorisés qui l'utilisent (détection du spectre) [2].

La RC peut augmenter l'efficacité spectrale, car elle permet à SU de partager opportuniste le spectre avec PU.

Une RC est définie aussi comme une radio qui peut changer ses paramètres d'émetteur en fonction de l'interaction avec l'environnement dans lequel il opère [20]. La RC a la capacité (capacité cognitive) de sentir et de rassembler informations (telles que la fréquence de

transmission, qualité, puissance, modulation, etc.) des environs environnement [20] ainsi que la capacité d'adapter rapidement le paramètres opérationnels, pour une performance optimale, selon l'information recueillie [6].

D'après la Définition de Joseph Mitola III « la Radio cognitive (CR) est une radio capable d'analyser l'environnement (en tant que canaux et utilisateurs), apprenant et prédisant la manière la plus appropriée et efficace d'utiliser le spectre disponible et d'adapter son fonctionnement paramètres » [1].

Aussi elle est définie comme la technique prometteuse pour l'utilisation efficace du spectre qui doit surveiller l'activité dynamique dans le spectre du primaire et adapter la transmission des SU pour une meilleure allocation des ressources spectrales. Ce système radio qui permet, à l'aide d'une radio logicielle, de définir ou de modifier les paramètres de fonctionnement de la fréquence radio d'un nœud réseau (téléphone sans fil ou un point d'accès sans fil) et de s'ajuste automatiquement pour maintenir les communications souhaitées - c'est comme si un opérateur formé «à l'intérieur» de la radio effectuait des réglages constants pour une performance maximale [3].

Un exemple très populaire de la radio opportuniste ou l'accès au spectre opportuniste dont le principe est la "réutilisation" temporelle, spatiale et géographique du spectre sous licence, comme le montre la **figure I.2** où un utilisateur secondaire "non autorisé" peut utiliser des fréquences sous licence. Qu'il n'interfère avec aucun (PU). De cette manière, l'efficacité de l'utilisation du spectre est considérablement améliorée [5].

Certes la RC est une nouvelle approche qui favorise une meilleure utilisation des bandes spectrales, mais elle requiert l'intervention d'une autre technologie qui est la radio logicielle (software defined radio) ou SDR.

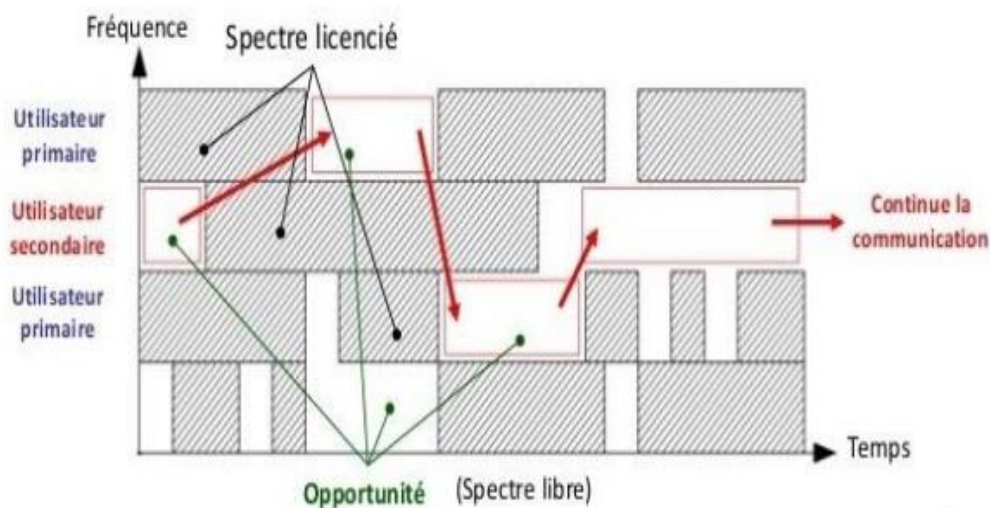


Figure I.2 : Exemple des trous spectraux [4]

I.3 RADIO LOGICIEL

Le SDR est un système de radiocommunication configurable utilisant des techniques de traitement numérique du signal sur des circuits numériques programmables. Sa flexibilité lui permet de s'adapter à différents protocoles de radiocommunication, et de répondre au besoin croissant de performance et d'interopérabilité entre systèmes [6].

Autrement dit que RL permette à un même système matériel d'exécuter plusieurs fonctions radios différentes. Pour y arriver, l'un des fondements du RL est de positionner la conversion entre les modes analogiques et numériques aussi proches que possible de l'antenne, aussi bien à l'émission qu'à la réception.

Grace à RL la dématérialisation complète de l'interface radio est possible. Elle participe à la tendance globale des circuits électroniques à devenir des circuits à haute densité d'intégration. L'évolution ultime de la radio logicielle est radio cognitive puis la radio intelligente [2].

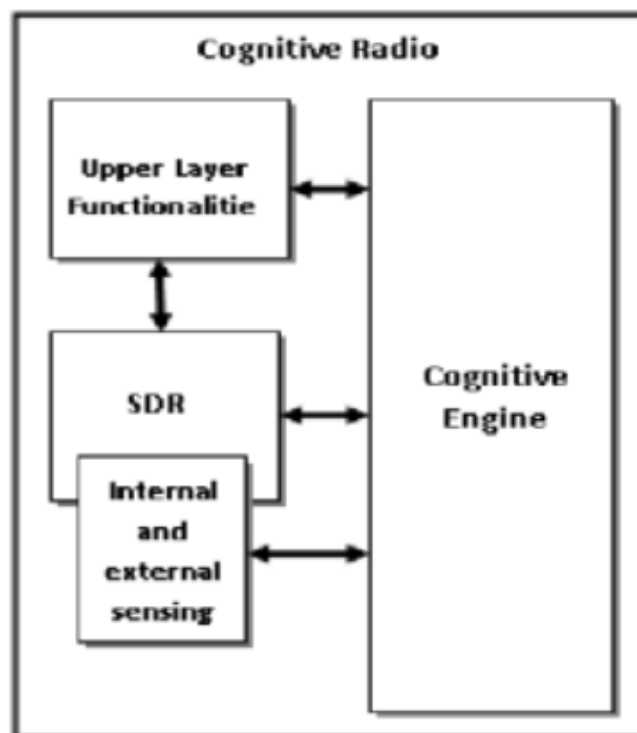


Figure I.3 : relation entre radio cognitive et radio logiciel [4]

I.4 Auto-configuration de la radio cognitive

La RC est capable de s'auto-configurer dynamiquement en ajustant ses paramètres de transmission en temps réel en n'important aucune modification à ses **composantes** matérielles. Cette caractéristique est la faculté majeure qui permet à la RC de s'ajuster à l'environnement dynamique où elle opère [9]. D'après le rapport de la FCC, les paramètres qui doivent être reconfigurables sont les suivants:

Fréquence d'opération

La RC est capable de changer de fréquence d'opération quand cette dernière devient indisponible. Le choix de la nouvelle fréquence est basé sur l'information collectée de l'environnement radio. La fréquence la plus appropriée est sélectionnée et la communication est réalisée sur cette nouvelle fréquence [6].

Modulation

La RC reconfigure son schéma de modulation selon les besoins l'utilisateur et selon les conditions de la bande spectrale. Dans le cas des applications telles que la voix sur IP, la RC doit sélectionner le schéma de modulation qui garantit la meilleure efficacité spectrale (débit le plus élevé). Dans le cas des applications qui ne supportent pas un taux d'erreur élevé, la RC doit sélectionner le schéma de modulation qui garantit le taux d'erreur le plus bas [6].

Puissance de transmission

Lors du changement de la qualité ou lors d'un déplacement de l'émetteur ou du récepteur. Les paramètres de transmission incluant la puissance de transmission changent de manière autonome. La nouvelle puissance de transmission est choisie selon les nouvelles contraintes que la RC doit respecter afin d'utiliser les BS d'une manière opportuniste. Le contrôleur de puissance est l'entité qui rend possible cet ajustement de puissance tout en s'assurant que les seuils d'interférence causée aux utilisateurs primaires ne soient pas dépassés dans l'environnement radio. Quand les besoins de la transmission sont satisfaits, et que cette dernière n'a plus besoin d'avantage de puissance, le contrôleur réduit la puissance pour permettre à d'autres usagers d'utiliser la même BS et diminuer l'interférence causée aux autres transmissions secondaires [6].

I.5 Architecture de la radio cognitive

Les six composantes fonctionnelles de l'architecture de la radio cognitive sont [7]:

1. L'interface de perception sensorielle de l'utilisateur comprend les fonctions de capture (sensing) et de perception, visuelles et acoustiques.
2. Les capteurs de l'environnement local (position, température, accélération, etc.).
3. Les applications système (les services médias indépendants comme un jeu en réseau).

4. Les fonctions de la radio logicielle restreinte (SDR) (qui incluent la détection RF et les applications radio de la SDR).
5. Les fonctions de la cognition (contrôle, planification, apprentissage).
6. Les fonctions locales effectrices (synthèse de parole, texte, graphiques, affichages multimédias). [2]

I.6 Architecture des réseaux radio cognitive

Une description détaillée de l'architecture des RRC est primordiale pour développer des protocoles efficaces de communication. Ces protocoles serviront par la suite à supporter les capacités cognitives [12].

Les éléments qui composent le RRC se décomposent en deux réseaux distincts : primaire et secondaire.

I.6.1 Le réseau primaire

Un utilisateur primaire

(Utilisateur licencié) est un utilisateur qui détient une licence pour opérer sur des bandes spectrales qui lui sont réservées. L'accès est contrôlé uniquement par ses stations de base et ne doit pas subir d'interférence extérieure nuisible. Les PU ne doivent subir aucune modification pour permettre la coexistence avec les utilisateurs ou réseaux de radios cognitives ou leurs stations de base [12].

Une station de base primaire

(Station de base licenciée) est une infra-structure fixe du réseau primaire qui possède une licence pour opérer sur la bande spectrale (par exemple les stations de base des systèmes cellulaires). Ces stations de bases sont conçues pour ne pas avoir à partager le spectre avec aucune entité extérieure du système[12].

I.6.2 Le réseau secondaire

(RCC, réseau à accès dynamique, ou bien réseau non licencié) est un réseau qui n'a pas de licence pour opérer sur la bande spectrale. Les réseaux secondaires sont déployés en mode infrastructure ou en mode ad-hoc [12], comme le montre **la Figure 1.4**. Ils se composent des éléments suivants:

Un utilisateur à radio cognitive

(Utilisateur non licencié ou utilisateur secondaire) n'a pas de licence pour transmettre sur la bande spectrale. Cependant, grâce aux fonctionnalités additionnelles dont ils disposent, ces

utilisateurs pourront partager la bande spectrale avec les utilisateurs primaires à condition de ne pas nuire leurs transmissions ou bien profiter de leur absence pour transmettre [12].

Une station de base secondaire

(Station de base non licenciée) est une infrastructure fixe avec des capacités cognitives. L'URC se connecte à la station de base secondaire pour accéder à d'autres réseaux ou services [12].

Un serveur spectral

(Spectrum server) est une entité du RRC qui sert à partager les ressources spectrales entre différents URC dans le même réseau. Ce serveur est connecté aux réseaux secondaires et agit comme un gestionnaire d'information spectrale [12].

Un courtier spectral

(Spectrum broker) est une entité du RRC qui partage les ressources spectrales entre différents RRC. Ce serveur est connecté à plusieurs RRC et agit comme un gestionnaire d'information spectrale [12].

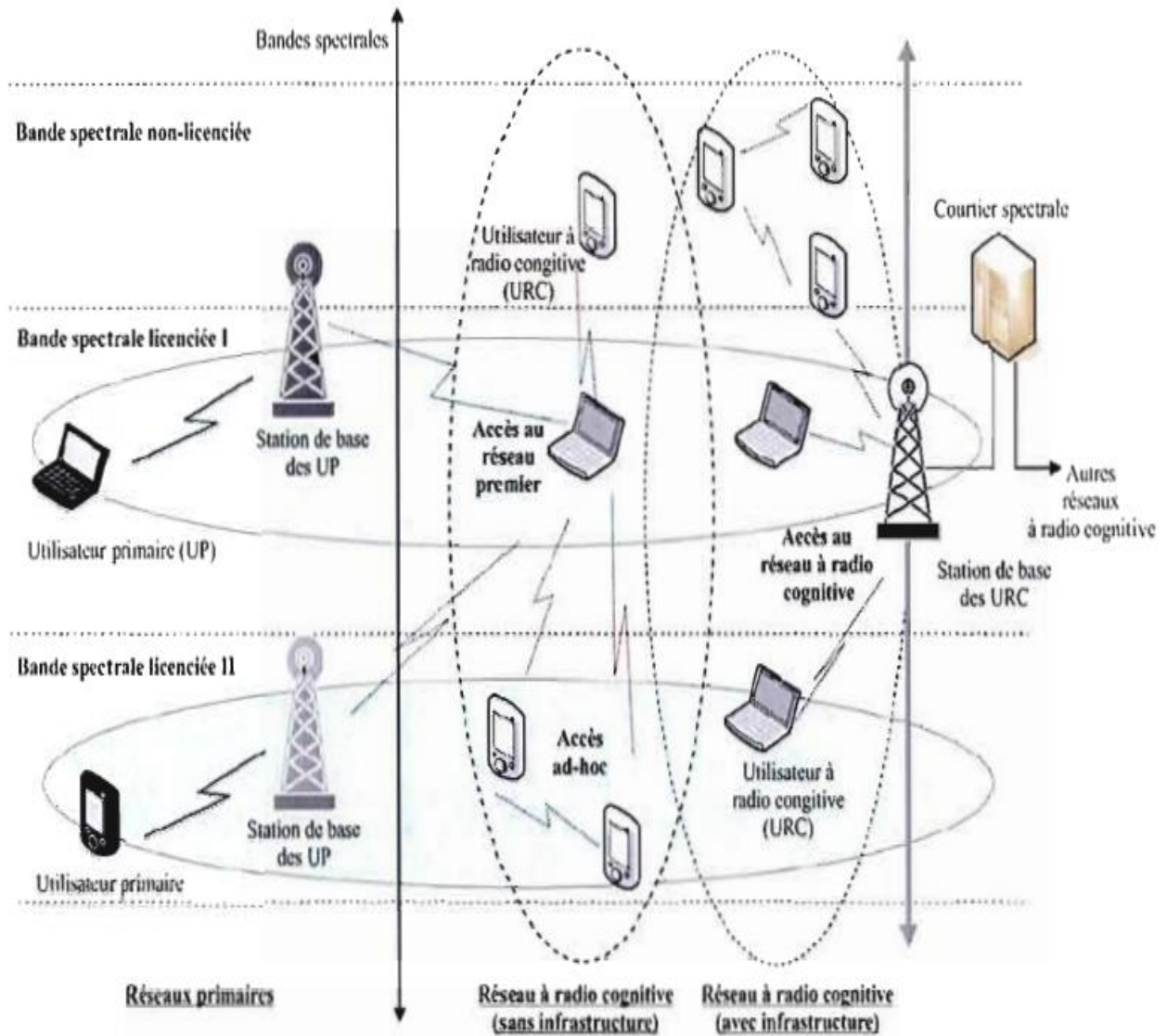


Figure I.4 : Coexistence entre deux types réseau : Primaire et secondaire [12]

I.7 Cycle de cognition

Le cycle cognitif consiste en différentes étapes comme indiqué dans la **figure I.5**. Cette figure décrit en détail ce cycle commençant par l'étape d'observation jusqu'à l'étape d'action afin de permettre à la radio cognitive d'interagir avec son environnement.

Les systèmes cognitifs observent, orientent, planifient, décident et agissent, tout en apprenant de leur environnement afin d'être plus efficaces au fil du temps. Les différentes étapes du cycle cognitif sont les suivantes :

- Observation : Extraire plusieurs informations à partir de l'environnement comme la fréquence radio, le type de données transmises (audio, vidéo, etc.), la position, etc.

- Orientation : Evaluer la situation et déterminer si elle est familière et réagir immédiatement, si nécessaire.
- Planification : Identifier les actions alternatives à prendre.
- Décision : Décider entre les actions candidates, en choisissant la meilleure d'entre elles.
- Action : Agir sur l'environnement en effectuant, par exemple, des modifications au niveau de la fréquence radio.
- Apprentissage automatique : S'informer à partir de l'expérience acquise à travers l'observation de l'environnement.

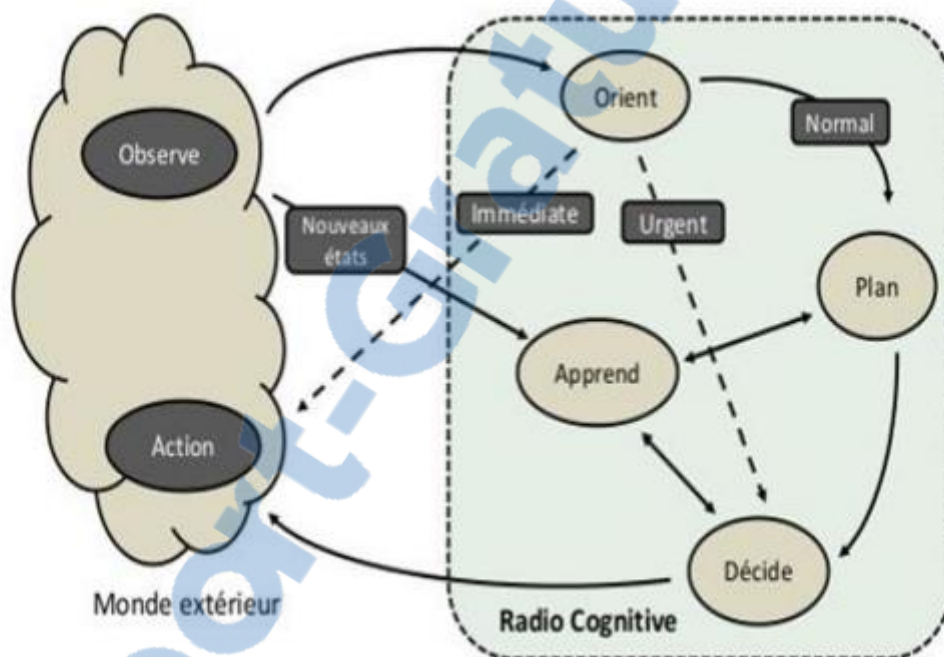


Figure I.5 : Le cycle de cognition de Mitola et Maguire. [4]

I.8 Fonctions RC

Gestion spectrale

Les bandes spectrales inutilisées ont des caractéristiques différentes les unes des autres. Ces caractéristiques sont la fréquence d'opération de la bande spectrale, le débit et le temps. Toutes ces informations changent au cours du temps vu la nature dynamique de l'environnement radio. C'est dans ce contexte que [1] ont présenté les nouvelles fonctions requises pour gérer les

ressources spectrales dans les RRC. Ces fonctions sont la détection spectrale, l'analyse spectrale et la décision spectrale.

Analyse spectrale

Elle permet de caractériser les différentes bandes spectrales en termes de fréquence d'opération, de débit, de temps et de l'activité de PU. Cette caractérisation sert à répondre aux exigences de l'URC. Des paramètres supplémentaires viennent compléter cette caractérisation, à savoir, le niveau d'interférence, le taux d'erreur du canal, le délai et le temps d'occupation de la bande spectrale par un URC [1].

Décision spectrale

après que toutes les bandes spectrales aient été catégorisées et classifiées, nous appliquons un ensemble de règles décisionnelles pour obtenir la ou les bandes spectrales les plus appropriées à la transmission en cours, en tenant compte des exigences de l'URC [1].

Mobilité du spectre

La mobilité spectrale est définie comme étant le processus de changement de fréquence d'opération par un URC. À ce moment-là, les deux URC (émetteur/récepteur) devront évacuer la BS et trouver une autre BS disponible pour continuer la communication. Dans ce contexte, le changement de fréquence (connu en anglais sous le nom de (hand-off) doit être redéfini pour mieux s'intégrer aux RRC. Les protocoles des différentes couches doivent s'adapter aux paramètres de la nouvelle fréquence d'opération à chaque fois que les URC changent de fréquence. L'objectif de la mobilité spectrale est d'assurer une transition fluide et rapide lors du changement de fréquence. Ceci est essentiel pour que les applications des URC subissent le moins de dégradation possible dans leurs performances durant la mobilité spectrale [1].

I.9 Conclusion

Les ressources radio et particulièrement le spectre, sont considérées comme une ressource précieuse, non pas à cause de leur indisponibilité mais parce qu'elles sont utilisées de manière inefficace. La philosophie principale de la technologie CR est donc de permettre aux utilisateurs sans licence d'accéder aux bandes dédiées aux utilisateurs sous licence (PU), mais sans entraîner de dégradation du service pour les titulaires de licence d'origine.

Chapitre II : Optimisation multi-objectifs, Décision Multicritères (topsis)



II. CHAPITRE II : Optimisation multi-objectifs, Décision Multicritères (topsis)

II.1 Introduction

Dans la plupart des problèmes du monde réel, il ne s'agit pas d'optimiser seulement un seul critère mais plutôt d'optimiser simultanément plusieurs critères et qui sont généralement conflictuels. Dans les problèmes de conception, par exemple, il faut le plus souvent trouver un compromis entre des besoins technologiques et des objectifs de coût. L'optimisation multi objective consiste donc à optimiser simultanément plusieurs fonctions. La notion de solution optimale unique dans l'optimisation uni-objective disparaît pour les problèmes d'optimisation multi objective au profit de la notion d'ensemble de solutions Pareto optimales.

II.2 Dominance de Pareto

II.2.1 Définition

L'optimisation multi-objective cherche à optimiser simultanément plusieurs critères souvent contradictoires. Il ne s'agit plus dans ce cas de trouver une solution optimale mais un ensemble de solutions représentant un compromis acceptable entre les différents objectifs contradictoires et connus comme l'ensemble des solutions Pareto optimales [16].

Un problème d'optimisation combinatoire multi-objectif (PMO) peut être défini comme suit :

$$\left\{ \begin{array}{l} \text{"Optimiser"} \\ F(x) = (f_1(x), f_2(x) \dots f_k(x)) \end{array} \right. \quad \text{sous } x \in X$$

Où k est le nombre d'objectifs ($k \geq 2$), $x = (x_1, x_2, \dots, x_n)$ est le vecteur représentant les variables de décision, chacune des fonctions $f_i(x)$ est à optimiser $i = 1, \dots, k$, c'est-à-dire à maximiser et X représente l'ensemble des solutions réalisables.

- l'ensemble R^n qui contient X est dit espace de décision.

- l'ensemble R^k qui contient F est dit espace des critères ou espace des objectifs.

II.2.1 Concepts de bases

Lors de la résolution d'un problème d'optimisation multi-objectif, nous obtenons une multitude de solutions. Seul un nombre restreint de ces solutions va nous intéresser

Définition

Soient deux vecteurs critères $z, z' \in F(X)$. On dit que z domine z' si et seulement si $z \geq z'$ et $z \neq z'$ (i.e. $z_i \geq z'_i \forall i = 1, \dots, k$ et $z \geq z'$ pour au moins un indice i). Cela veut dire que z est au moins aussi bon que z' dans tous les objectifs et, z est strictement meilleur que z' dans au moins un objectif [15].

Dominance forte

Soient deux vecteurs critères $z, z' \in F(X)$. On dit que z domine fortement z' si et seulement si $z > z'$ et $z \neq z'$ (i.e. $z_i > z'_i \forall i = 1, \dots, k$). Cela veut dire que z est meilleur que z' sur tous les critères [15].

Définition (Pareto optimal)

Soit $x = [x_1, \dots, x_n]$ un vecteur de décision avec $x \in X$ (L'espace réalisable) x est dit Pareto optimal, s'il n'existe pas une solution y domine x . Une solution Pareto optimal appelée aussi solution efficace, non inférieure ou non dominée solution. Les solutions qui dominent les autres mais ne se dominent pas entre elles sont appelées solutions optimales au sens de Pareto (ou solutions non dominées), elles forment le front de paréto [15].

Définition

Le front (frontière) de Pareto est l'ensemble des solutions Pareto optimales qui sont composées des points, ne sont dominés par aucun autre. Le front de Pareto appelé aussi surface de compromise ou l'ensemble des solutions efficaces. **La figure II.1**, représente le Front du Pareto pour un problème de maximisation de deux fonctions objective (f_1, f_2).

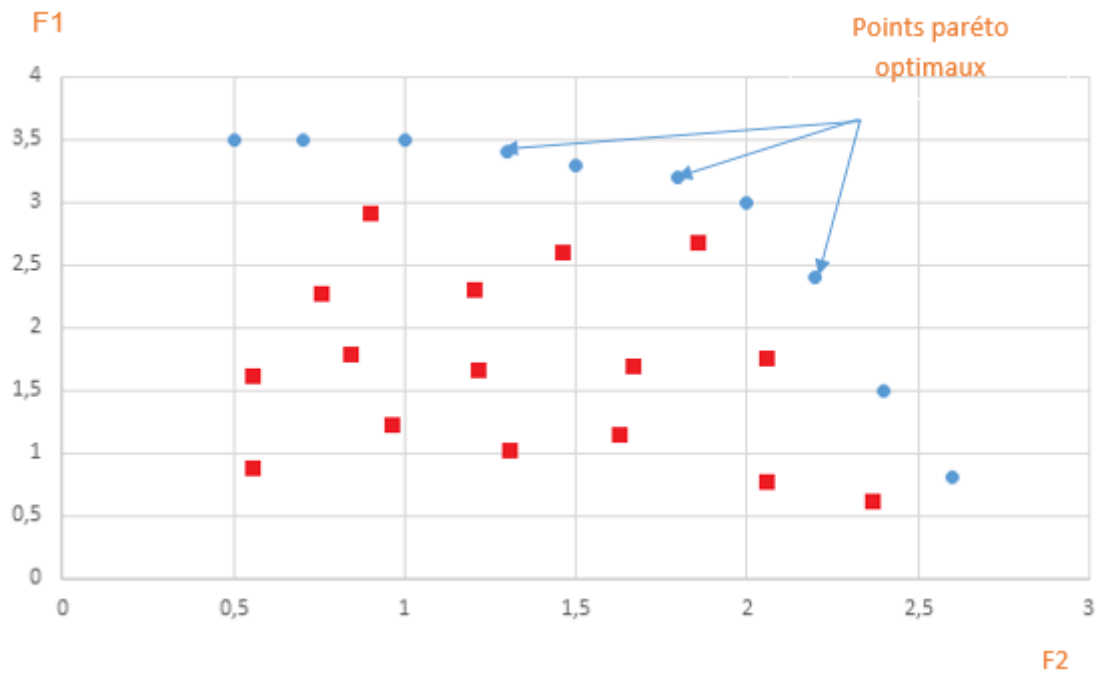


Figure II.1 : Front Pareto : ensemble solution non dominée.

Dans les problèmes multi-objectifs, l'optimum n'est plus une simple valeur comme pour les problèmes mono-objectif, mais un ensemble de points, appelé l'ensemble des meilleurs compromis ou le front Pareto [16].

II.2.3 Points particuliers

D'après **figure II.2** et en vue d'avoir certains points de références permettant de discuter de l'intérêt des solutions trouvées, des points particuliers ont été définis dans l'espace objectif. Ces points peuvent représenter des solutions réalisables ou non.

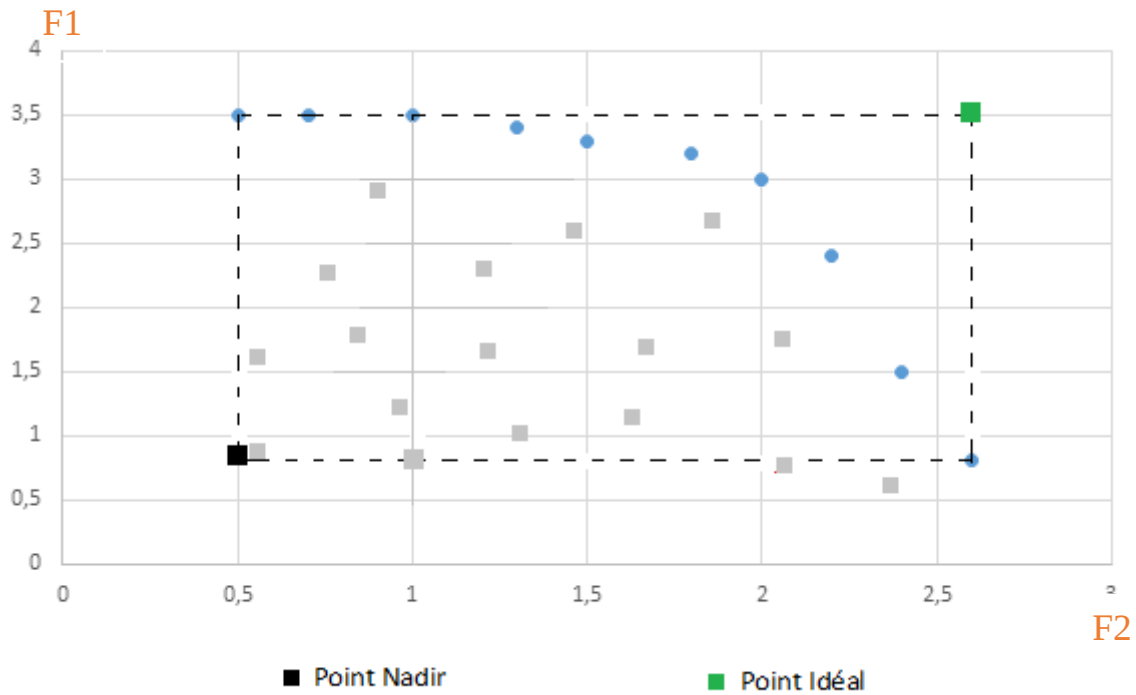


Figure II.2 : Points particuliers : point Idéal, point nadir.

Tout d'abord, le point idéal est le point qui a comme valeur pour chaque objectif la valeur optimale de l'objectif considéré. Ce point ne correspond pas à une solution réalisable malheureusement, cela sous-entendrait que les objectifs ne sont pas contradictoires et qu'une solution optimisant un objectif, optimise simultanément tous les autres, ce qui ramènerait le problème à un problème ayant une seule solution Pareto optimale.

Enfin le point Nadir qui correspond aux pires valeurs obtenues par chaque fonction objectif, lorsque l'on restreint l'espace des solutions à la surface de compromis.

II.3 Méthode de décision Multicritères

La méthode TOPSIS est une des méthodes les plus utilisées d'aide multicritère à la décision. Elle est basée sur le choix d'une solution qui se rapproche le plus de la solution idéale (meilleure sur tous les critères) et s'éloigne le plus de la pire solution (qui dégrade tous les critères). Elle compare un ensemble d'alternatives par l'identification des poids de chaque critère, la normalisation des scores de chaque critère et le calcul de la distance euclidienne entre chaque alternative et l'alternative idéale qui a le meilleur score dans chaque critère. Les m poids des critères sont compris entre 0 et 1 et définis par le vecteur $W = (W_1, W_2, \dots, W_m)$. Certains critères ont un effet positif (critères de bénéfice) si une bonne alternative obtient un score élevé sur ce critère. D'autres critères ont un effet négatif (critère de coût) si une bonne alternative par rapport à ce critère a un score peu élevé [4].

Cette méthode est capable de trier les solutions du front Pareto en considérant les solutions comme des alternatives et les valeurs des fonctions objectif de chaque solution comme des critères.

II.3.1 Matrice de décision

Pour n alternative (Option) et m Critères, Nous avons une matrice $M = (D_{ij})$ matrice de $n \times m$.

$$M = \begin{bmatrix} D_{11} & \dots & D_{1m} \\ \vdots & \ddots & \vdots \\ D_{n1} & \dots & D_{nm} \end{bmatrix}$$

II.3.2 Algorithme TOPSIS

Les différentes étapes de TOPSIS sont les suivantes :

Etape 1 : Normalisation des performances.

$$N_{jk} = \frac{D_{jk}}{\sqrt{\sum_{j=1}^n D_{jk}^2}} ; j = 1, 2, \dots, n ; k = 1, 2, \dots, m$$

Où n est le nombre d'alternatives, m est le nombre de critères et D_{ij} est la valeur associée à l'alternative j selon le critère k .

Etape 2 : Calcul du produit des performances normalisées par les valeurs des pondérations relatives aux critères

$$V_{jk} = N_{jk} * W_k; j = 1, 2, \dots, n; k = 1, 2, \dots, m$$

Où W_k représente la valeur de la pondération du critère k et V_{jk} est la matrice des performances normalisée et pondérée.

Etape 3: détermination des solutions idéales (V^+) et anti-idéales (V^-).

$$V^+ = \{(\max_j V_{jk} \mid k \in K), (\min_j V_{jk} \mid k \in K') \mid j = 1, 2, \dots, n; k = 1, 2, \dots, m\}$$

$$V^- = \{(\min_j V_{jk} \mid k \in K), (\max_j V_{jk} \mid k \in K') \mid j = 1, 2, \dots, n; k = 1, 2, \dots, m\}$$

Où K représente l'ensemble de critères de bénéfice et K' représente l'ensemble de critères de coût.

Etape 4 : Calcul de la distance euclidienne par rapport aux solutions V^+ et V^-

$$S_j^+ = \sqrt{\sum_{k=1}^m (V_{jk} - V^+)^2}; j = 1, 2, \dots, n; k = 1, 2, \dots, m$$

$$S_j^- = \sqrt{\sum_{k=1}^m (V_{jk} - V^-)^2}; j = 1, 2, \dots, n; k = 1, 2, \dots, m$$

Etape 5: Calcul de l'index de similarité à la solution idéale.

$$P_j^* = \frac{S_j^-}{S_j^+ + S_j^-}; j = 1, 2, \dots, n; 0 \leq P_j^* \leq 1$$

Etape 6 : Ordre de préférence

Choisir l'action ayant le plus grand index de similarité (problématique de choix).
Ranger les actions par ordre décroissant des index de similarité (problématique de rangement).

II.4 Conclusion

Nous avons présenté dans ce chapitre les principaux concepts de l'optimisation multi-objectif sur les méthodes d'optimisation de problèmes multi-objectifs. Ce concept est appelé L'approche Pareto, qui adopte un point de vue plus global en prenant en compte l'ensemble des critères et en utilisant la notion de dominance au sens de Pareto. Nous avons vu aussi le concept de décision multi critère et sa résolution avec la méthode topsis qui servira comme outil d'aide à la sélection de la meilleure solution parmi les solutions non dominées obtenues par les approches multicritères.

Chapitre III :
méthodes de résolution exacte et
Approchées

III. Chapitre III: Méthode de résolution exacte et approchées

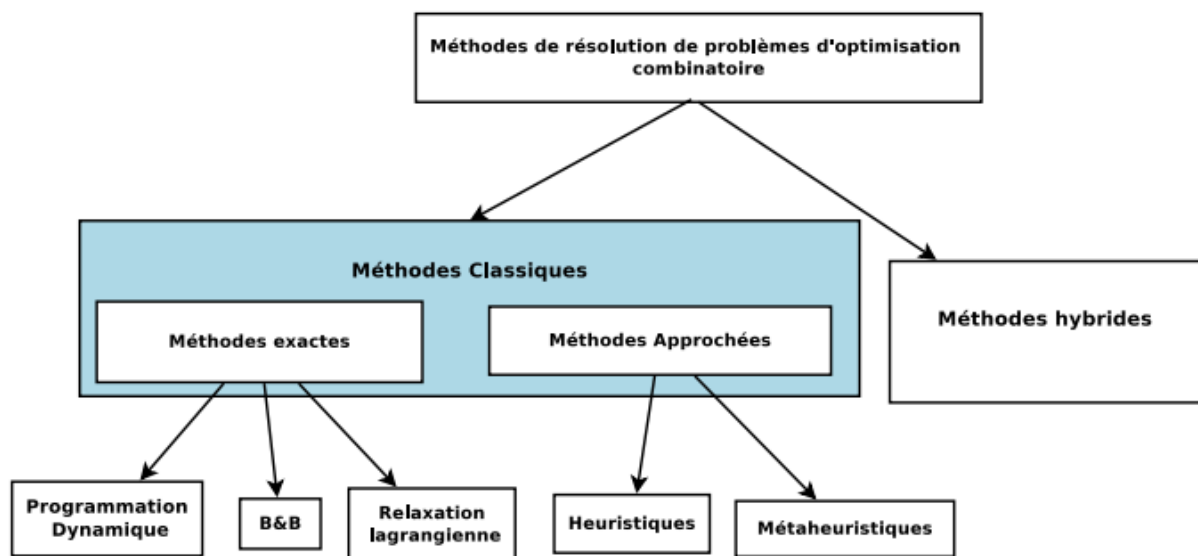
III.1 Introduction

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle et en informatique. De nombreuses applications pouvant être modélisées sous la forme d'un problème d'optimisation combinatoire (POC) telles que le problème du voyageur de commerce, l'ordonnancement de tâches, le problème de la coloration de graphes, etc. POC comprend un ensemble fini de solutions, où chaque solution doit satisfaire un ensemble de contraintes relatives à la nature du problème, et une fonction objectif pour évaluer chaque solution trouvée. La solution optimale est celle dont la valeur de l'objectif est la plus petite (resp. grande) dans le cas de minimisation (resp. maximisation) parmi l'ensemble de solutions.

La résolution des problèmes combinatoires consiste à trouver la meilleure solution, définie comme la solution globalement optimale ou un optimum global. La résolution des (POC) est assez délicate puisque le nombre fini de solutions réalisables croît généralement avec la taille du problème, ainsi que sa complexité. Cela a poussé les chercheurs à développer de nombreuses méthodes de résolution en recherche opérationnelle et en (IA) [17].

Il existe un très grand nombre de méthodes de résolutions des problèmes d'optimisation combinatoire et ils sont classés en deux grandes catégories:

Les méthodes exactes et les méthodes approchées



Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable et rencontrent généralement des difficultés face aux applications de taille importante

Alors, si les méthodes de résolution exactes permettent d'obtenir une solution dont l'optimalité est garantie, dans certaines situations, on peut cependant chercher des solutions de bonne qualité, sans garantie d'optimalité, mais au profit d'un temps de calcul plus réduit. Pour cela, On applique des méthodes appelées métaheuristiques, adaptées à chaque problème traité, avec cependant l'inconvénient de ne disposer en retour d'aucune information sur la qualité des solutions obtenues.

Les heuristiques ou les métaheuristiques exploitent généralement des processus aléatoires dans l'exploration de l'espace de recherche pour faire face à l'explosion combinatoire engendré par l'utilisation des méthodes exactes. En plus de cette base stochastique, les métaheuristiques sont le plus souvent itératives, ainsi le même processus de recherche est répété lors de la résolution.

III.2 Les méthodes de résolution exactes

Ces méthodes explorent l'espace de recherche de manière exhaustive et permettent de trouver la solution exacte d'un problème d'optimisation combinatoire, ces méthodes donnent une garantie de trouver la solution optimale pour une instance de taille finie dans un temps limité et de prouver son optimalité [17].

Nous présentons dans cette sous-section deux méthodes sur lesquelles se basent un grand nombre d'algorithmes pour la résolution exacte des (POC) et sur les qu'elles nous avons utilisé durant ce travail :

La méthode de programmation dynamique et la méthode de séparation et d'évaluation (Branch and Bound).

III.2.1 Programmation dynamique

La programmation dynamique est une technique algorithmique pour optimiser des sommes de fonctions monotones croissantes sous contraintes. Elle a été désignée par ce terme pour la première fois dans les années 1940 par Richard Bellman. Elle s'applique à des problèmes d'optimisation dont la fonction objective se décrit comme $\ll$ la somme de fonctions monotones croissantes des sources $\gg$.

Autrement, la technique de programmation dynamique est une approche générale qui apparaît comme un outil utile pour résoudre divers problèmes en optimisation combinatoire. L'idée de base qui se trouve derrière cette technique a été introduite par Bellman. Cette approche consiste

à décomposer un problème en sous-problèmes plus simples, ensuite résoudre ces sous-problèmes et combiner leurs solutions afin de trouver une solution globale.

La programmation dynamique s'appuie sur un principe simple :

Toute solution optimale s'appuie elle-même sur des sous problèmes résolus localement de façon optimale. Concrètement, cela signifie que l'on va pouvoir déduire la solution optimale d'un problème en combinant des solutions optimales d'une série de sous problèmes. Les solutions des problèmes sont étudiées 'de bas en haut', c'est -à-dire que nous calculons les solutions des sous problèmes les plus petits pour ensuite déduire petit à petit la solution de l'ensemble.

III.2.1.1 Programmation dynamique parallèle

Le parallélisme constitue une autre approche afin d'accélérer la résolution de problèmes d'optimisation combinatoire. L'utilisation des threads semble particulièrement intéressante afin de diminuer les temps de résolution de manière économique.

Notre travail, s'est concentré sur la mise en œuvre parallèle via les threads de la méthode de Programmation dynamique.

Le nombre de threads par bloc est restreint par les ressources mémoire limitées d'un processeur.

III.2.1 Méthode Branch and Bound

Présentation de l'algorithme

La **figure II.2** représente l'algorithme de séparation et évaluation, plus connu sous son appellation anglaise Branch and Bound (B&B) [18], repose sur une méthode arborescente de recherche d'une solution optimale par séparations et évaluations, en représentant les états solutions par un arbre d'états, avec des nœuds, et des feuilles.

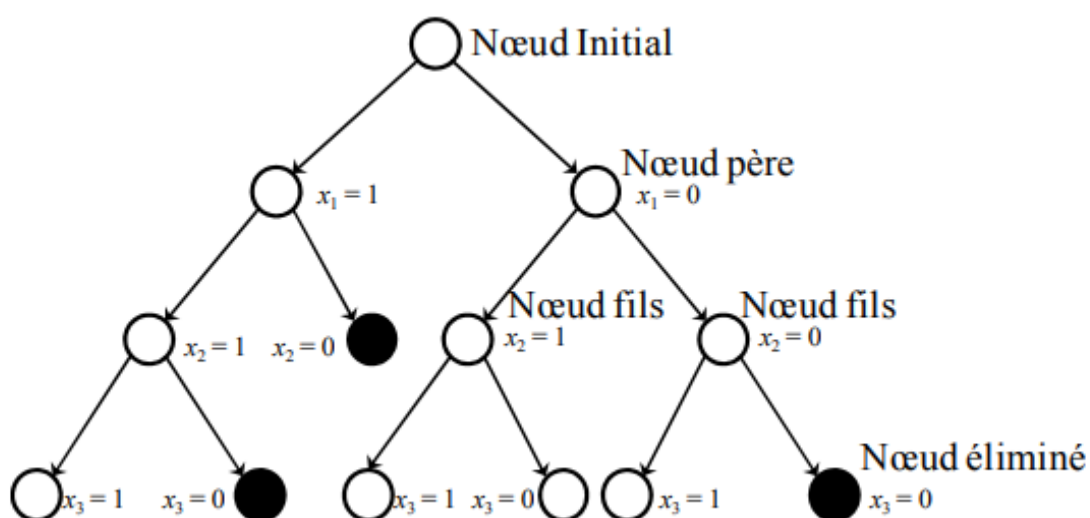


Figure II.2 : Principe algorithme séparation et évaluation (B&B)

Chaque sous-problème créé au cours de l'exploration est symbolisé par un nœud de l'arbre (ou sommet), le nœud racine représentant le problème initial.

Les branches de l'arbre symbolisent le processus de séparation. Elles représentent la relation entre les nœuds. La méthode tente d'explorer intelligemment l'ensemble des solutions admissibles en éliminant de l'espace de recherche tous les sous-ensembles de solutions qui ne peuvent pas fournir une solution optimale.

L'algorithme est basé sur trois axes principaux:

- L'évaluation
- La séparation
- La stratégie de parcours.

Principe d'évaluation

Le principe d'évaluation a pour objectif de connaître la qualité des nœuds à traiter.

La méthode de Branch and Bound utilise deux types de bornes.

- une borne inférieure de la fonction d'utilité du problème initial,
- une borne supérieure de la fonction d'utilité.

La connaissance d'une borne inférieure du problème et d'une borne supérieure de la fonction d'utilité de chaque sous-problème permet de stopper l'exploration d'un sous-ensemble de solutions qui ne sont pas candidates à l'optimalité : si pour un sous-problème la borne supérieure est plus petite que la borne inférieure du problème, l'exploration du sous-ensemble correspondant est inutile.

D'autre part, chacune des solutions partielles potentielles possède une borne supérieure et une autre inférieure. Ces dernières sont utilisées pour couper quelques branches de l'arbre et ainsi éviter d'explorer tout l'arbre.

Le principe de séparation

Le principe de séparation consiste à diviser le problème en un certain nombre de sous problèmes qui ont chacun leur ensemble de solutions réalisables. En résolvant tous les sous problèmes et en prenant la meilleure solution trouvée, on est assuré d'avoir résolu le problème initial. Ce principe de séparation est appliqué de manière récursive à chacun des sous-ensembles tant que celui-ci contient plusieurs solutions.

Remarque : La procédure de séparation d'un ensemble s'arrête lorsqu'une des conditions suivantes est vérifiée :

- on connaît la meilleure solution de l'ensemble.
- on connaît une solution meilleure que toutes celles de l'ensemble .
- on sait que l'ensemble ne contient aucune solution admissible.

Stratégie de parcours

La stratégie de parcours est la règle qui permet de choisir le prochain sommet à séparer parmi l'ensemble des sommets de l'arborescence.

Parmi les stratégies de parcours les plus connues, on peut citer :

➤ La profondeur d'abord

L'exploration privilégie les sous-problèmes obtenus par le plus grand nombre de séparations appliquées au problème de départ, c'est-à-dire aux sommets les plus éloignés de la racine (de profondeur la plus élevée).

L'obtention rapide d'une solution admissible (pour les problèmes où il est difficile d'obtenir une heuristique de bonne qualité) et le peu de place mémoire nécessaire en sont les avantages. L'inconvénient est l'exploration de sous-ensembles qui peuvent s'avérer peu prometteurs à l'obtention d'une solution optimale.

➤ La largeur d'abord

Cette stratégie favorise les sous-problèmes obtenus par le moins de séparations du problème de départ, c'est-à-dire les sommets les plus proches de la racine (de profondeur la moins élevée).

➤ Le meilleur d'abord

Cette stratégie favorise l'exploration des sous-problèmes possédant les plus petites bornes inférieures. Elle dirige la recherche là où la probabilité de trouver une meilleure solution est la plus grande.

III.3 Méthode de résolution Approchées

En optimisation combinatoire, une heuristique est un algorithme approché qui permet d'identifier en temps polynomial au moins une solution réalisable rapide, pas obligatoirement optimale. L'usage d'une heuristique est efficace pour calculer une solution approchée d'un problème et ainsi accélérer le processus de résolution exacte. Généralement une heuristique est conçue pour un problème particulier, on s'appuyant sur sa structure propre sans offrir aucune garantie quant à la qualité de la solution calculée [19].

Métaheuristiques

Face aux difficultés rencontrées par les heuristiques pour avoir une solution réalisable de bonne qualité pour des problèmes d'optimisation difficiles, les métaheuristiques ont fait leur apparition. Ces algorithmes sont plus complets et complexes qu'une simple heuristique, et permettent généralement d'obtenir une solution de très bonne qualité pour des problèmes issus des domaines de la recherche opérationnelle dont on ne connaît pas de méthodes efficaces pour les traiter ou bien quand la résolution du problème nécessite un temps élevé ou une grande mémoire de stockage.

Le rapport entre le temps d'exécution et la qualité de la solution trouvée d'une métaheuristique reste alors dans la majorité des cas très intéressant par rapport aux différents types d'approches de résolution. La plupart des métaheuristiques utilisent des processus aléatoires et itératifs comme moyens de rassembler de l'information, d'explorer l'espace de recherche et de faire face à des problèmes comme l'explosion combinatoire. Plusieurs d'entre elles sont souvent inspirées par des systèmes naturels dans de nombreux domaines tels que la biologie (algorithmes évolutionnaires et génétiques) la physique (recuit simulé), et aussi l'éthologie (algorithmes de colonies de fourmis). Un des enjeux de la conception des métaheuristiques est donc de faciliter le choix d'une méthode et le réglage des paramètres pour les adapter à un problème donné.

Les métaheuristiques peuvent être classées de nombreuses façons. On peut distinguer celles qui travaillent avec une population de solutions de celles qui ne manipulent qu'une seule solution à la fois.

Nous présentons dans cette sous-section l'algorithme génétique.

Algorithme génétique

Les algorithmes génétiques sont des algorithmes d'optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils ont été adaptés à l'optimisation par John Holland [19] également les travaux de David Goldberg ont largement contribué à les enrichir [19].

Le vocabulaire utilisé est le même que celui de la théorie de l'évolution et de la génétique, on emploie le terme individu (solution potentielle), population (ensemble de solutions), génotype (une représentation de la solution), gène (une partie du génotype), parent, enfant, reproduction, croisement, mutation, génération, etc. Leur fonctionnement est extrêmement simple, on part d'une population de solutions potentielles (chromosomes) initiales, arbitrairement choisies.

On évalue leur performance (Fitness) relative. Sur la base de ces performances on crée une nouvelle population de solutions potentielles en utilisant des opérateurs évolutionnaires simples:

la sélection, le croisement et la mutation. Quelques individus se reproduisent, d'autres disparaissent et seuls les individus les mieux adaptés sont supposés survivre. On recommence ce cycle jusqu'à ce qu'on trouve une solution satisfaisante. En effet, l'héritage génétique à travers les générations permet à la population d'être adaptée et donc répondre au critère d'optimisation, la **figure III.3** illustre les principales étapes d'un algorithme génétique.

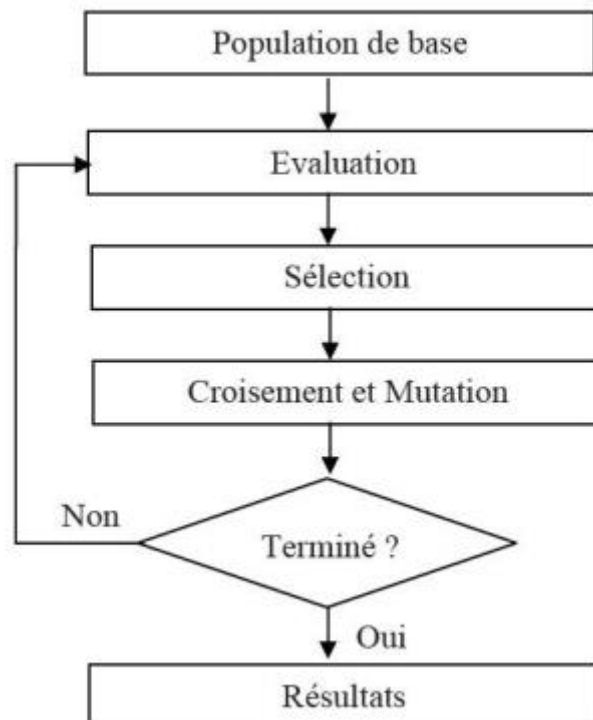


Figure III.3 : Organigramme Algorithme génétique.

La mise en œuvre d'un algorithme génétique nécessite :

a) Le codage des données

La première étape est de définir et coder convenablement le problème. Cette étape associe à chaque point de l'espace de recherche une structure de données spécifique, appelée génotype ou ensemble de chromosomes, qui caractérisera chaque individu de la population.

Le codage de chaque individu en séquence est essentielle dans l'élaboration d'un algorithme génétique dont dépend notamment l'implémentation des opérateurs de transformations. Ainsi, cette phase détermine la structure de données qui sera utilisée pour coder le génotype des individus de la population. Le codage doit donc être adapté au problème traité.

Plusieurs types de codages sont utilisés dans la littérature, les premiers résultats théoriques sur les algorithmes génétiques ont opté pour un codage par une séquence binaire de longueur fixe à travers la notion de schéma [19]. L'efficacité de l'algorithme génétique dépend donc du choix convenable du type de codage.

b) Génération de la population initiale

La génération de la population initiale, c'est-à-dire le choix des dispositifs de départ que nous allons faire évoluer, ce choix de la population initiale d'individus conditionne fortement la rapidité de l'algorithme. Néanmoins, une initialisation aléatoire est plus simple à réaliser. Les valeurs des gènes sont tirées au hasard selon une distribution uniforme. Toutefois, il peut être utile de guider la génération initiale vers des sous domaines intéressants de l'espace de recherche.

Par exemple lors d'une recherche d'optimalité dans un problème d'optimisation sous contraintes, il est préférable de produire des éléments satisfaisant les contraintes. La population initiale doit être suffisamment diversifiée et de taille assez importante pour que la recherche puisse parcourir l'espace d'état dans un temps limité.

c) Fonction d'évaluation

L'évaluation de la Fitness est généralement l'étape dans laquelle on mesure la performance de chaque individu. Pour pouvoir juger la qualité d'un individu et ainsi le comparer aux autres, il faut établir une mesure commune d'évaluation. Aucune règle n'existe pour définir cette fonction, son calcul peut ainsi être quelconque, que ce soit une simple équation ou une fonction affine. La manière la plus simple est de poser la fonction d'adaptation comme la formalisation du critère d'optimisation.

d) La Sélection

La sélection permet d'identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais, pendant le passage d'une génération à une autre, ce processus est basé sur la performance de l'individu. L'opérateur de sélection doit être conçu pour donner également une chance aux mauvais éléments, car ces éléments peuvent, par croisement ou mutation, engendrer une descendance pertinente par rapport au critère d'optimisation.

Il existe différentes techniques de sélection, nous avons opté pour une sélection par élitisme.

Élitisme : Cette méthode de sélection permet de favoriser les meilleurs individus de la population. Ce sont donc les individus les plus prometteurs qui vont participer à l'amélioration de notre population.

On peut constater que cette méthode induisait une convergence prématurée de l'algorithme

e) Croisement

L'opérateur de croisement favorise l'exploration de l'espace de recherche et enrichit la diversité de la population en manipulant la structure des chromosomes, le croisement fait avec deux parents et génère deux enfants, en espérant qu'un des deux enfants au moins héritera de bons gènes des deux parents et sera mieux adapté qu'eux.

Il existe plusieurs méthodes de croisement par exemple le croisement en un point, ou en multiples points **Figure III.4**.

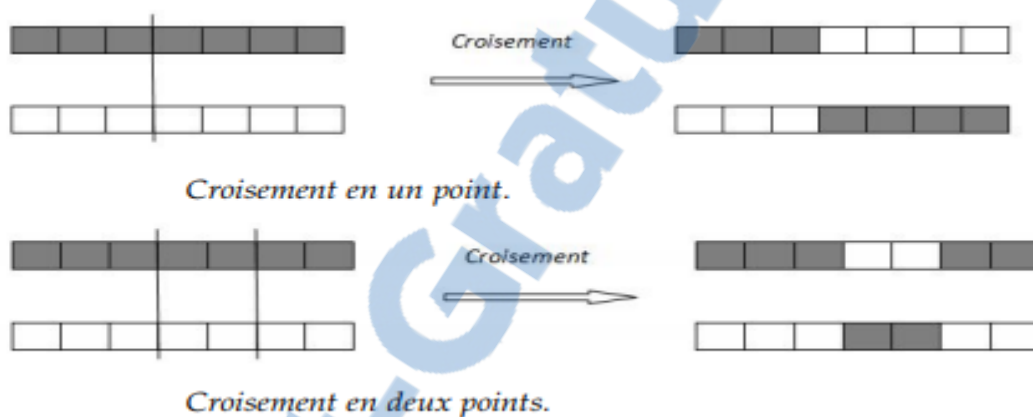


Figure III.4 : exemple de croisement à point multiple.

f) Mutation

Comme montre la **figure III.5** l'opérateur de mutation est un processus où un changement mineur du code génétique appliqué à un individu pour introduire de la diversité et ainsi d'éviter de tomber dans des optimums locaux. L'efficacité des algorithmes génétiques dépend fortement du réglage des différents paramètres caractérisant La probabilité de croisement et La probabilité de mutation dépendent directement de la nature du problème et de sa complexité, et leurs choix doit représenter un compromis entre la qualité des solutions et le temps d'exécution.

La probabilité de croisement p_c est liée à la forme de la fonction d'évaluation. Son choix est en général heuristique. Plus sa valeur est élevée, plus la population subit des changements importants. La probabilité de mutation p_m est généralement faible puisqu'un taux élevé risque de conduire vers un optimum local. En revanche, une probabilité faible permet d'assurer une bonne exploration de l'espace de recherche sans perturber la convergence.

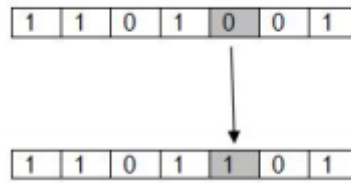


Figure III.5 : exemple de Mutation a changement minimum.

III.4 Conclusion

Les trois chapitres précédant avaient pour objectif de présenter dans un premier temps les principales définitions nécessaires à la radio cognitive présentation des problèmes d'optimisation combinatoire multi-objectif et quelques méthodes de résolution exacte et approché, dans le chapitre suivant nous allons focaliser notre étude sur un cas particulier de ces problèmes, le problème de trouver les meilleurs canaux libres en termes de prix et de qualité cote PU ainsi pour assurer continuellement une très bonne qualité pour les utilisateur opportunistes.

Chapitre IV: Implémentation et expérimentation

IV. Chapitre IV: Implémentation et expérimentation

IV.1 Introduction

L'idée clé de la décision sur le spectre dans les RRC est la sélection de la meilleure BF disponible pour satisfaire les exigences des (SU), sans interférer avec la transmission des (PU). Les RRC préfèrent utiliser des métaheuristiques pour résoudre problème d'allocation des ressources spectral afin de satisfaire le plus grand nombre des (SU). Les approches exactes sont très vite limitées, et des métaheuristiques dédiées à la résolution de ces problèmes ont été mises au point, mais la difficulté de ce type de problèmes qui appartient à la classe des problèmes NP-difficile encourage l'utilisation de méthodes d'optimisation parallélisée pour cela, nous avons choisi la programmation dynamique et algorithme de séparation et évaluation (B&B) comme méthodes exactes, et une métaheuristiques qui est AG comme méthodes approchées, afin d'établir une étude comparative selon différents critères, tels que le temps de traitement, le gain obtenu.

Note : Les méthodes exactes et la métaheuristique ont été implémentées sous le langage JAVA, sur un ordinateur fonctionnant sous le système d'exploitation Windows 8 caractérisé par **1 CPU Intel I3** avec **4 cœurs** et une mémoire vive de **8 GB DDR3**.

IV.2 Fonction objectif

La fonction de décision du spectre sélectionne une bande spectral basée sur les exigences de qualité de service (QoS) des SU, telles que la qualité, le nombre de canaux demandé, ainsi que le temps d'allocation. Dans notre cas cette fonction a pour but de maximiser le coût « bénéfice pour l'utilisateur primaire » et aussi maximiser la qualité et de minimiser le temps de traitement « temps de réponse aux utilisateurs secondaire ». Soit les paramètres suivants :

- n : le nombre des SU.
- m : le nombre de canaux libres coté PU.
- W : tableau de taille n .
- $W[i]$ est le nombre de canaux demandés par $SU(i)$.
- T_1 : tableau de taille n ,
- $T_1[i]$ représente le prix proposé pour $W[i]$ par $SU(i)$.
- T_2 : tableau de taille n ,

- $T_2[i]$ représente la qualité du $W[i]$ demandé par $SU(i)$.
- Les deux fonctions à optimiser est : $\text{Max } \sum_{i=1}^n T_1[i]$, $\text{Max } \sum_{i=1}^n T_2[i]$
- La contrainte à respecter est : $\sum_{i=1}^n W[i] \leq m$

Les paramètres utilisés sont comme suit :

1. $n = 1200$.
2. $m = 1500$.

Les formules des tableaux W et T_1, T_2 sont :

3. $W[i] = \text{Random}(i) \quad 10 \leq i \leq 1200$.
4. $T_1[i] = \text{Random}(i) \quad 500 \leq i \leq 9000$.
5. $T_2[i] = \text{Random}(i) \quad 100 \leq i \leq 8024$.

Cependant ces paramètres précédents seront toujours utilisés dans toutes les algorithmes qui suivent pour effectuer l'étude comparative.

IV.3 Algorithme génétique

Les AGs ont été largement utilisés dans la communauté multi-objectif. Ils sont très appropriés pour résoudre des problèmes multi-objectifs grâce à l'utilisation d'une population de solutions. Les AGs peuvent chercher plusieurs solutions Pareto-optimales dans la même exécution.

Comme tout algorithme évolutionnaire, on a un ensemble de paramètres à ajuster pour bien appliquer la programmation sur un problème. Ces paramètres permettent d'aider l'algorithme à converger d'une manière efficace et rapide vers la bonne solution.

- Notre algorithme génétique a été initialisé avec :
- Population de 40 individus.
- Un taux de croisement égal à 0.8 de type Point Fixe
- Un taux de mutation de 0.05
- Une sélection par tournoi.

Afin de parvenir à de meilleurs résultats la condition d'arrêt pour la métaheuristique était le fait d'atteindre un certain nombre de génération (1000, 2000, 3000, 4000). En effet, suite à plusieurs expériences, nous avons décidé de choisir le chiffre de 1000 génération comme critère d'arrêt pendant la comparaison avec les deux méthodes exactes, car il s'est avéré à ce niveau de critère d'arrêt notre algorithme génétique montre des résultats raisonnable en temps exécution.

Algorithme général

algorithme génétique
Initialisation de la population
Evaluation de fonctions objectives
Calcul de l'efficacité
For I = 1 to MaxIter
<i>Sélection aléatoire,</i>
Sélection proportionnelle à l'efficacité
Croisement,
Mutation,
<i>Evaluation des fonctions objectives,</i>
Calcul de l'efficacité
End For

Cependant, le choix d'une population avec une taille importante et un critère d'arrêt trop grand peut augmenter considérablement le temps de calcul, et si la taille de la population est trop petite, il y aura une convergence rapide car l'algorithme n'a pas un grand aperçu sur l'espace de recherche Voir **figure IV.1**.

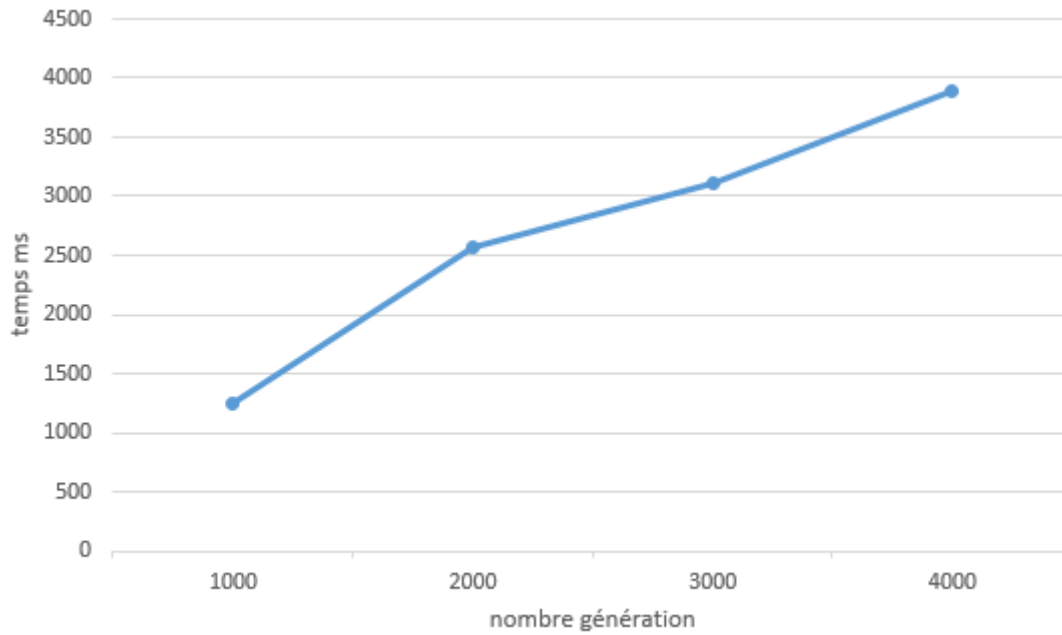


Figure IV.1 : résultat en termes temps exécution (ms)

Nous constatons un temps de traitement raisonnable pour l'algorithme génétique aux dessous des 3000 générations testées et qui se stabilise entre la barre de (1s et 2.5 s) mais après avoir franchis la barre des 3000 générations, le graphe va s'accroître à cause de la perte du temps.

Nous traitons par suite Les coûts et la qualité avec la moyenne de 10 exécutions par génération. Les différentes exécutions obtenues sont illustrés à travers le la **figure VI.2**.

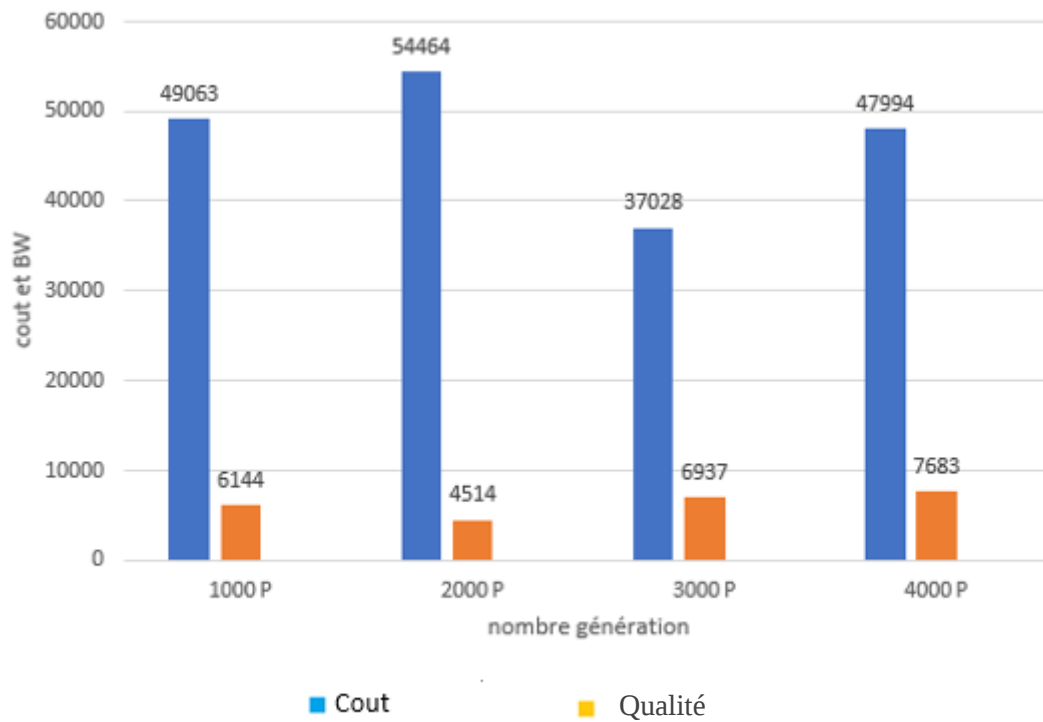


Figure IV.2 : résultat en termes de gain (prix, qualité) (génétique)

D'après le graphe **figure IV.2** Nous remarquons que le résultat en terme du gain de la métaheuristique génétique et quasiment fixes et n'est pas liées forcément avec le nombre de génération calculé, leurs valeurs sont presque inchangées par rapport nombre de génération voir tableau qui récapitule les résultats trouvés **tableau IV.1**.

Nbr génération	Couts	qualité	Temps (ms)
1000	49063.0	6144.0	1423
2000	54464	4514	2638
3000	37028	6937	3285
4000	47994	7683	3812

Tableau IV.1 : tableau récapitulatif des résultats (génétique)

Par la suite Nous avons choisis le nombre de la génération qui offre un meilleur temps d'exécution (1000 génération).

IV.4 Programmation dynamique

Pour que le moteur cognitif puisse résoudre le problème de l'allocation du spectre, nous proposons d'utiliser la programmation dynamique qui est une technique algorithmique pour optimiser des sommes de fonctions monotones croissantes sous contrainte. Cette technique s'applique à des problèmes d'optimisation dont la fonction objective se décrit comme « la somme de fonctions monotones croissantes des ressources ».

L'idée de base est d'éviter de calculer deux fois la même chose, généralement en utilisant une table de résultats déjà calculés, remplie au fur et à mesure qu'on résout les sous problèmes.

Le scénario présenté ressemble beaucoup au problème du sac à dos (Knapsack problème) multi objectifs et pour résoudre ce genre de problèmes nous définissons une table, à chaque élément correspondra la solution d'un et d'un seul problème intermédiaire. Il faut donc qu'on puisse définir chaque sous problème qui sera traité au cours du calcul, ensuite il faut remplir cette table.

Algorithme dynamique

Algorithme programmation dynamique

Sortie : Ensemble solution optimal

```

Begin
Pour i=0 -> n
    | Pour j=0 -> m
        | si (j >= W[i-1]) alors
            | T[i][j]= maxPareto(T [i-1][j], T[i-1][j-W[i-1]] + T[i-1])
        | sinon
            | T[i][j]= T[i-1][j]
        Finsi
    Fin pour
Fin pour

Fin

```

Cependant, il est nécessaire de spécifier un poids à chaque objectif pour que TOPSIS puisse classer les différentes solutions. Ainsi, dans ce cas d'étude, la technique de TOPSIS a classé la solution se trouvant sur la quatrième ligne comme étant la solution préférée, et cela en se basant sur les poids associés aux trois objectifs.

Pour cela, nous nous basons sur les mêmes données que celles de l'exemple précédent. Nous implémentons le programme avec 2 objectifs : le coût total, la qualité, avec respectivement les poids 0.5 et 0.5.

Les résultats obtenus sont présentés dans le **Tableau VI.2**. Il est clair que sans l'utilisation de TOPSIS, il est très difficile voire impossible que le SU puisse prendre une décision précise en se basant sur les résultats obtenus et récoltés des différents PU, tout en garantissant le meilleur compromis entre les différents objectifs.

Topsis	Prix	Qualité
1	97340	7940
2	96479	8420
3	90699	8434

Tableau IV.2 : résultats obtenus triés avec TOPSIS

Après que nous avons formulé l'algorithme général de la programmation dynamique adaptative sous une forme concurrente, il est cependant facile de le paralléliser. En effet, il suffit pour cela d'exécuter la boucle principale de manière indépendante par plusieurs threads qui ne communiquent entre eux que par l'intermédiaire de la mémoire.

Programation dynamique parallèle

Dans le cas de z threads (voir **figure IV.3**), l'idée est d'exécuter le premier thread sur la première ligne, le deuxième thread sur la deuxième ligne, le troisième thread sur la troisième ligne, etc. Mais comme le traitement de la ligne x_1 occupé par le thread numéro y_1 dépend du résultat de la ligne x et le thread y qui les précèdent. Alors Dans ce cas Le thread y_2 sera lancé sur la deuxième ligne lorsque le thread qui le précède a traité (ligne / z) cellules de la première ligne de la matrice (z ici est le nombre de threads). Le troisième thread sera lancé sur la troisième ligne lorsque le premier thread a traité $(2 * \text{ligne} / z)$ cellules de la première ligne et (ligne / z) cellules pour le deuxième thread.

Le quatrième thread sera lancé sur la quatrième ligne lorsque le premier thread a traité $(3 * \text{ligne} / z)$ cellules de la première ligne. Ensuite, le premier thread continuera à fonctionner sur la $(z+1)$ ème ligne. Le deuxième thread continuera à fonctionner sur la $(z+2)$ ème ligne, et ainsi de suite jusqu'à ce que toute la matrice soit traitée.

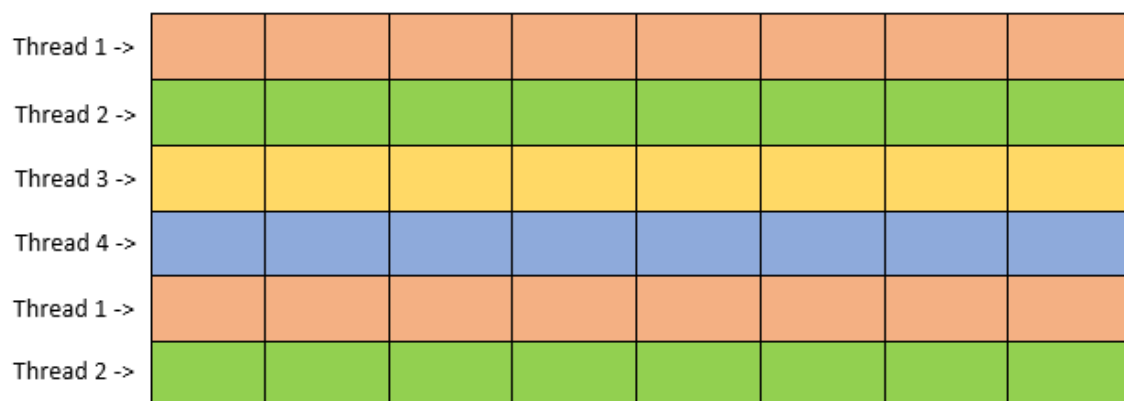


Figure IV.3 : principe des threads pour prg dynamique

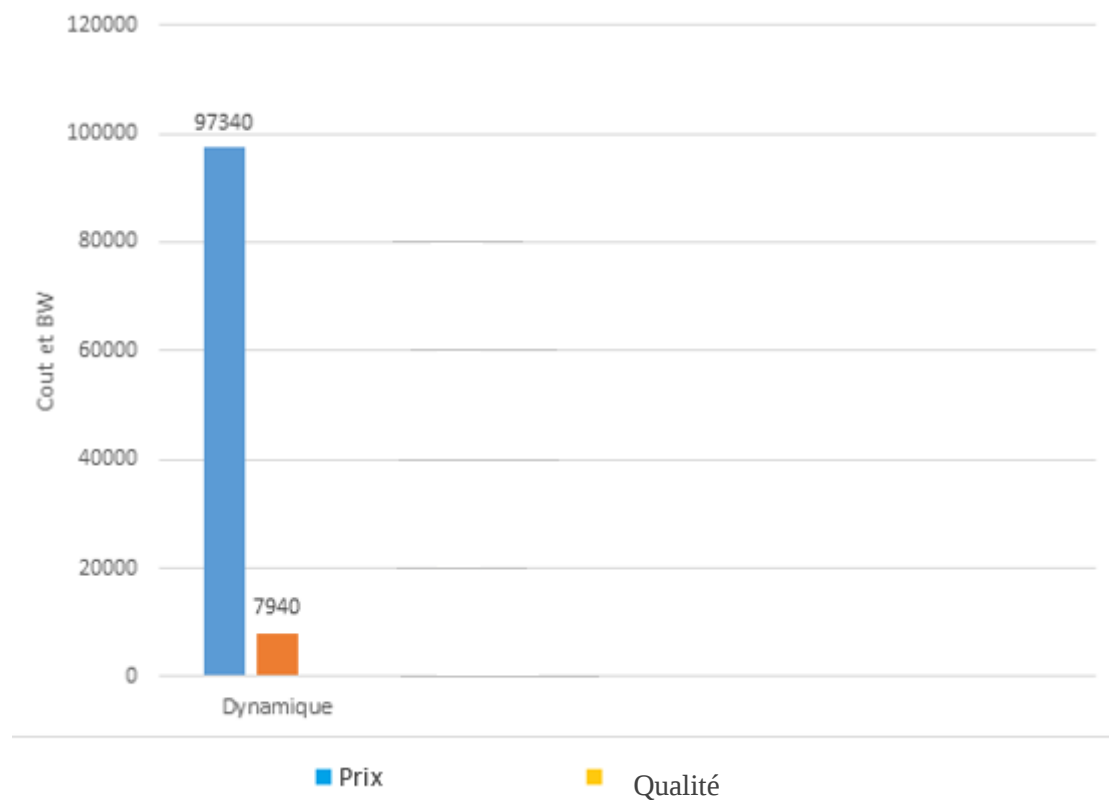


Figure IV.4 : résultats en termes de gain (prix, qualité) dynamique

Algorithme programmation dynamique parallèle
Sortie : Ensemble solution optimal

Public run ()

Debut:

si (ligne ==1) alors

| dynamique(1,1,m/nbthread)

| T thread=new T(2, m, C, W, nbthread)

Pour k=1 -> nbthread-2 faire

| dynamique(1,k*m/nbthread+1,(k+1)*m/nbthread)

| thread =new T(k+2, m, C, W, nbthread)

| thread.start()

Fin pour

dynamique(1,(nbthread-1)*m/nbthread+1,m)

ligne= ligne +nbthread

Tanque(ligne <=n) faire

| dynamique(ligne,1,m)

| ligne = ligne +nbthread

Fin tanque

Sinon

dynamique(ligne,1,m)

ligne = ligne +nbthread

Tanque(ligne <=n) faire

dynamique(ligne,1,m)

ligne = ligne +nbthread

Fin tanque

Fin si Fin

Le graphe **figure IV.5** représente les résultats obtenues avec une moyenne de 10 exécutions a été considérée pour chaque nombre de thread (2, 4, 8, 16, 32) Voir **figure IV.5**.

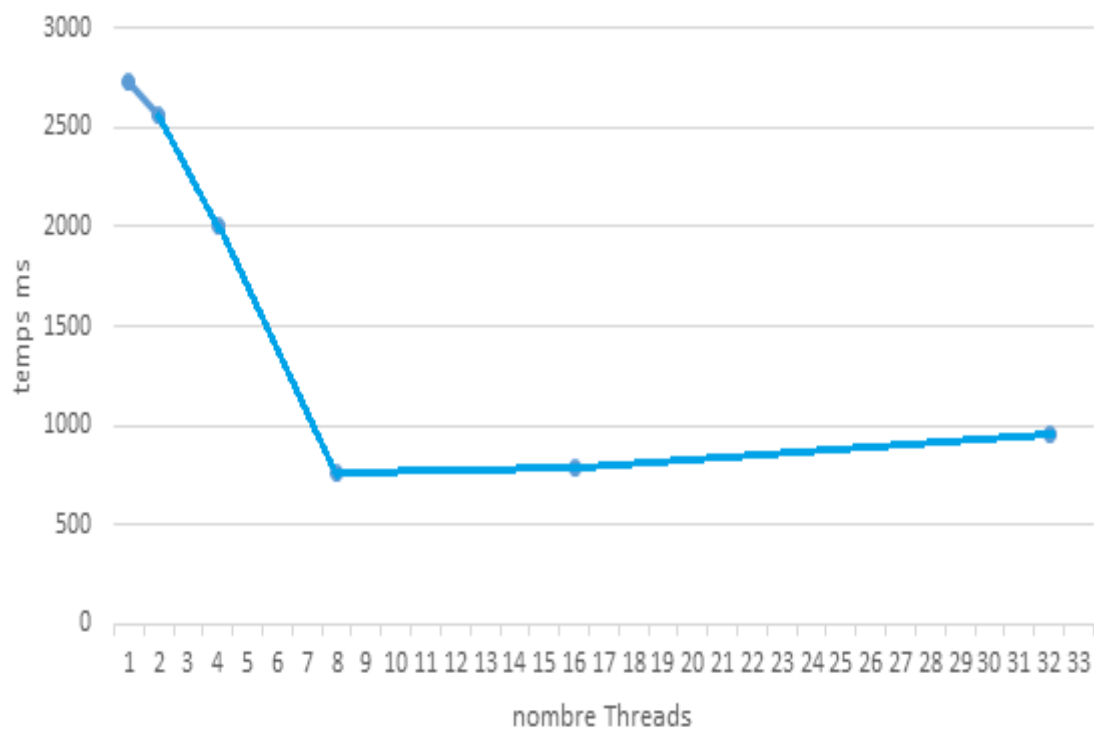


Figure IV.5 : Temps exécution par rapport aux nombre de threads

Le tableau suivant résume les résultats obtenus en terme temps exécution (**tableau IV.3**).

Nombre Threads	1	2	4	8	16	32
Temps exécution (ms)	2736	2512	2000	769	789	985

Tableau IV.3 : temps exécution programmation dynamique parallèle

Le graphe suivant **figure IV.6** présente une Comparaison entre AG et la programmation dynamique en terme temps exécution

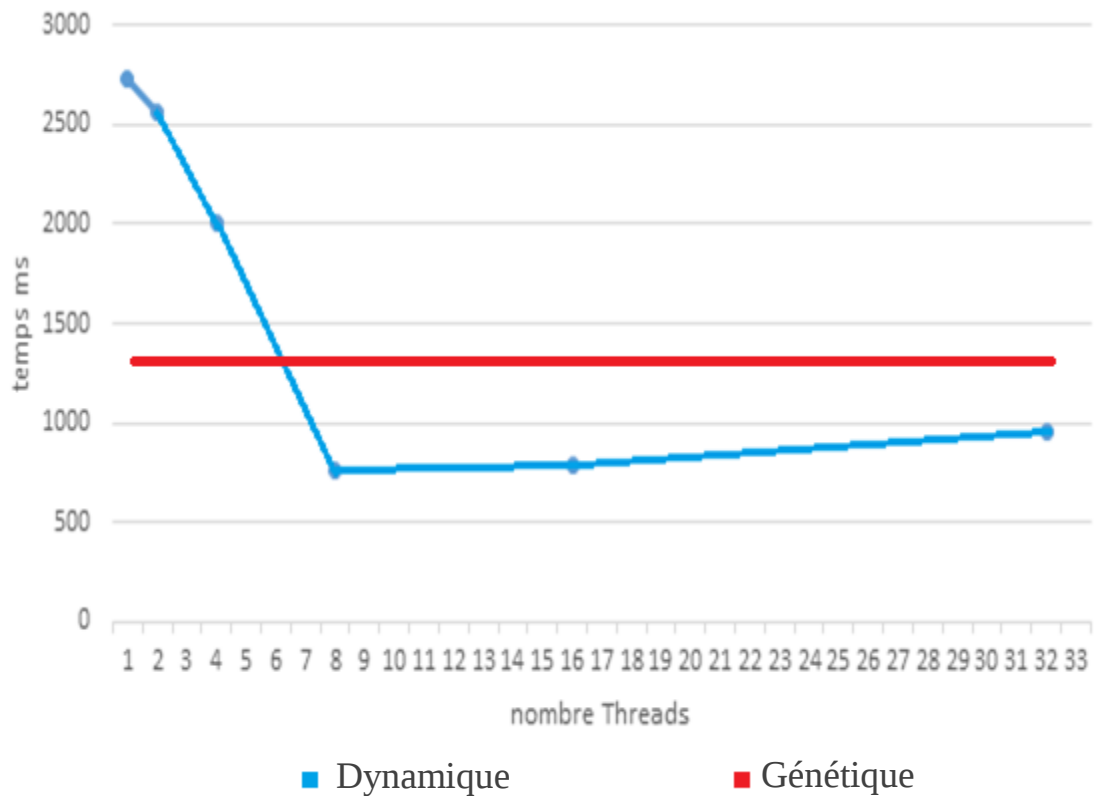


Figure IV.6 : Comparaison temps exécution (dynamique, génétique)

D'après le graphe précédent **figure IV.6** nous remarquons que le temps de traitement pour la programmation dynamique va diminuer entre 4 et 8 threads pour se stabiliser après avoir

franchis la barre des 8 threads, notez bien qu'une fois le nombre de threads dépasse 8, le graphe va s'accroître à cause de la perte du temps lors de la communication entre les threads.

On peut dire que la programmation dynamique offre de meilleurs temps d'exécution lorsqu'on utilise plus de 4 threads. En combinant la programmation dynamique (méthode exacte) et les threads, on obtient des résultats remarquablement supérieurs à l'algorithme génétique (métaheuristique) non seulement en termes de prix et qualité mais aussi en terme de temps d'exécution.

IV.5 Algorithme Branch and Bound

Généralement, les SUs et les PUs négocient leur accord par l'intermédiaire d'un CPU sur une base de multiples critères tels que le profit, le nombre de canaux, et le nombre de canaux disponible, prix. Cette multitude de paramètres nous a conduits à introduire une autre méthode exacte qui est basée sur la technique de séparation et d'évaluation.

Nous proposons un algorithme basé sur la procédure de séparation et d'évaluation progressive (Branch and Bound) une version modifiée reprenant la méthode de B&B avec une garantie de l'optimalité des solutions obtenues. Dans l'approche proposée.

Le principe du fonctionnement d'un algorithme B&B pour notre problème est défini comme suit : On choisit une variable parmi les variables non instanciées et on crée autant de nœuds fils que le nombre de valeurs de la variable. Pour chaque valeur, on calcule une borne inférieure et une borne supérieure. Si cette borne supérieure est inférieure à la borne inférieure maximale connue, on arrête alors l'exploration du sous-arbre engendré par cette valeur et on passe à une autre valeur.

Dans le cas contraire, il existe deux possibilités: soit on est à une feuille et dans ce cas, on met à jour la borne inférieure, soit on continue l'exploration de l'arbre de recherche. On peut remarquer que cette méthode est fortement dépendante de l'efficacité du calcul de la borne inférieure.

Le calcul des bornes (upper bound et lower bound) nous permet d'encadrer la valeur de la solution optimale. La plupart des bornes supérieures pour le problème se basent sur le tri des objets par ordre croissant d'efficacité.

Pour chaque nœud de l'arborescence, et pour un élément j pris dans l'ordre prédéfini, on développe deux branches: la première branche, $x_j = 1$ correspondant à l'élément j mis dans le sac c.-à-d. qu'il est pris en considération et la deuxième branche $x_j = 0$ correspond à l'élément j qui n'est pas mis dans le sac c.-à-d. en solution.

Autrement dit, Dans le cas d'un nœud fils correspondant à $x = 1$, la borne supérieure ainsi que la borne inférieure ne changent pas. Sinon Dans le cas d'un nœud fils correspondant à $x = 0$, on

calcule le nouvel indice de la variable de base Par conséquent, w et V_1 et V_2 sont mises à jour et les deux nouvelles bornes UB et LB, seront déduites.

Algorithme branch & bound
Sortie : solution optimal

Begin

Tree = { }

LB_max=null

Node root = { $v_1=0$, $v_2=0$, $w=0$, Finish=false}

Tree.add(root)

Sort the Nodes according to the formula : $(v_1+v_2)/W$

While(! All_Node_Finish())

 | choose node with high Upper bound

 | generate the childrens of node

 | remove node from tree

 | Add children to tree

 | Update best value of LB_max

 | For $i=0$ to tree.size()

 | if (node.Upper bound < LB_max) then

 | remove node(i)

 | end if

 | end for

end while

End

Dans ce travail, nous sommes arrivés à synchroniser entre seulement 4 threads l'idée est d'utiliser les threads pour l'évaluation en parallèle des différentes bornes à chaque étape de la méthode de Branch and Bound. L'étape de calcul de borne est ensuite lancée, en parallèle, pour évaluer les bornes supérieures et inférieures des nouveaux nœuds. On en tire alors la meilleure borne supérieure.

La borne inférieure Le du problème peut être obtenue à partir d'une solution admissible du problème. Par exemple, on peut avoir une borne inférieure intéressante en sélectionnant tous

les nœuds d'indice inférieur à l'indice de la variable en train de parcourir, et sachant que les nœuds sont triés par ordre croissant de ratio d'importance:

$$\frac{v_1 + v'_1}{w_1} > \frac{v_2 + v'_2}{w_2} > \frac{v_1 + v'_1}{w_3} > \dots > \frac{v_n + v'_n}{w_n} .$$

Enfin, L'étape d'élagage est lancée pour éliminer les nœuds «non prometteurs». Si pour un nœud e , nous avons une borne supérieure qui vérifie l'inégalité suivante : $UB(e) \leq LB_{max}$, alors ce nœud est supprimé de la liste. Dans le cas contraire, le nœud est conservé dans la liste.

Les étapes décrites ci-dessus représentent une itération de la méthode de Branch and Bound. On réitère ces étapes tant que :

- La liste des nœuds n'est pas vide.
- On n'a pas considéré tous les articles du problème réduit. La solution du problème est alors obtenue par la borne Lower.

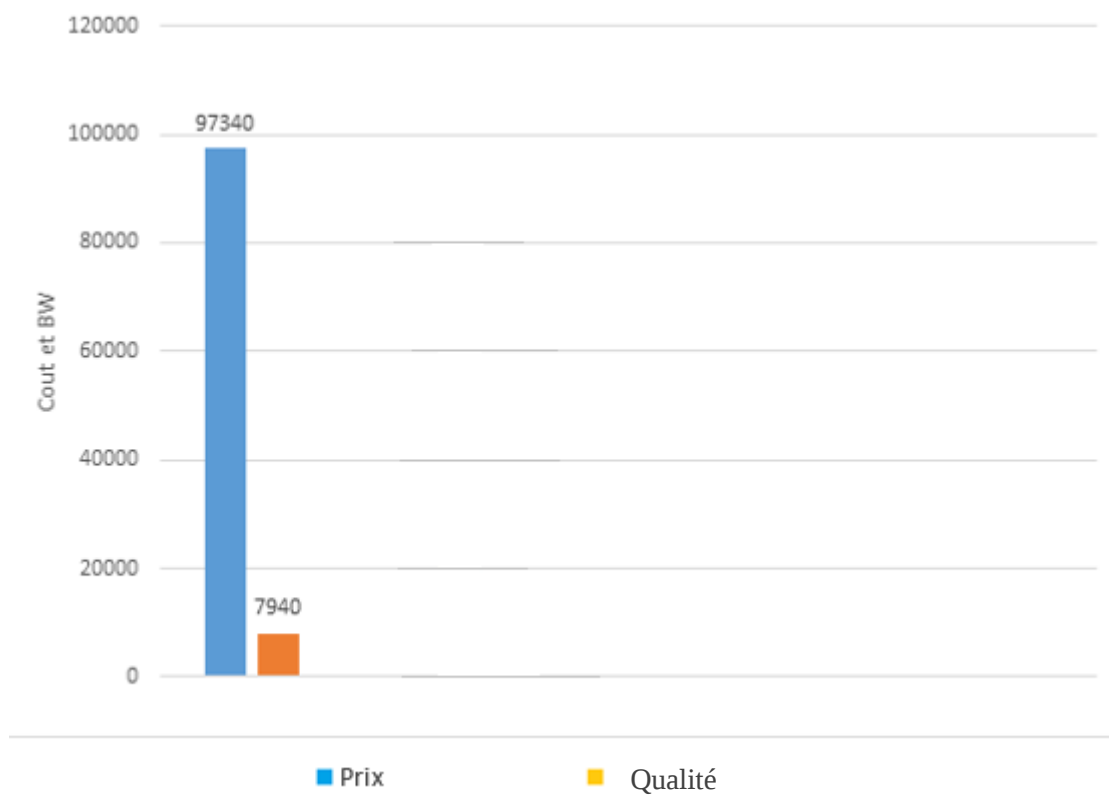


Figure IV.7 : résultats en termes de gain (prix, qualité) B&B

L'expérimentation précédente prouve les mêmes résultats en termes de gain. Donc, d'après les résultats, nous pouvons dire que l'algorithme B&B offre le même résultat que la programmation dynamique c'est-à-dire qu'en cette dernière on obtient des résultats optimaux en termes de gain indépendamment du nombre de threads combinant, c-à-d Cette valeur de gain obtenu par la programmation dynamique est carrément indépendante du nombre de threads utilisés, Mais qui ont montré leurs bénéfique en terme temps exécution.

Le tableau suivant récapitule les résultats trouvé.

prix	Qualité	Nbr threads	Temps exécution
97340	7940	4	742 ms

Tableau IV.4 : temps exécution B&B parallèle

Reste à comparer les résultats avec la métaheuristique.

IV.6 Comparaison des approches

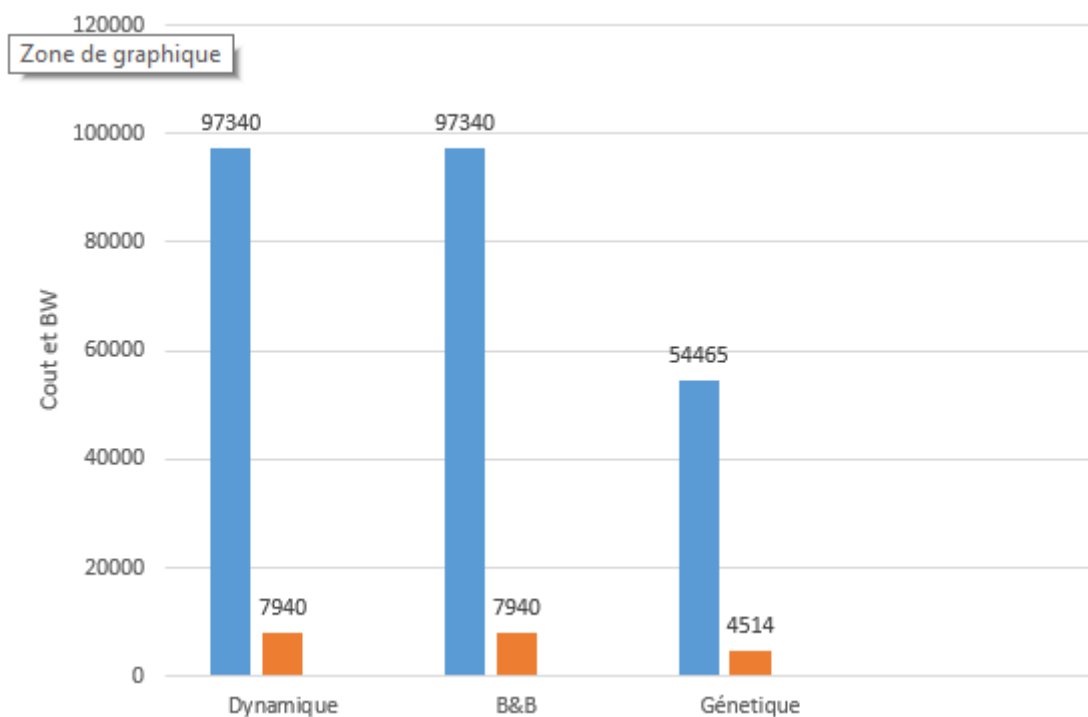


Figure IV.8: comparaison en termes de gain (prix, qualité)
(Dynamique, B&B, Génétique)

Algorithme	Programmation dynamique	B&B	AG
Temps exécution (ms)	769	742	1423

Tableau IV.5 : comparaison temps exécution

Les expériences montrent l'importance d'utiliser des méthodes exactes, aussi l'implémentation des threads pour ces méthodes est très bénéfique car elle permet d'obtenir le meilleur gain dans un temps raisonnable. Cette comparaison nous a permis de prouver que le parallélisme de la programmation dynamique et B&B abouti à de meilleurs résultats par rapport à l'algorithme génétique, on a remarqué aussi que la méthode B&B offre des résultats intéressantes par apport à l'approche dynamique.

IV.7 Conclusion

Afin de répondre à l'objectif de concevoir un algorithme d'optimisation multi objectif efficace, en temps de calculs (raisonnable), en qualité de solutions produites et permettant de traiter des problèmes de taille raisonnable, nous avons proposé dans ce chapitre deux algorithmes exacte pour résoudre le problème gestion spectral. Le parallélisme permet de réduire le temps d'attente des utilisateurs secondaires.

Un objectif principal était de comparer ces deux approches avec une métaheuristique, les expérimentations effectuées sur différents critères, tels que le temps de traitement, le coût obtenu ont prouvé leur efficacité.

Les résultats obtenus mettent en évidence le très bon comportement de notre proposition et montrent l'efficacité de parallélisme des approches exactes. Autrement dit, l'intérêt de l'utilisation des threads pour la méthode exacte utilisée pour l'optimisation des performances de la RC a déjà fait ses preuves.

V. Conclusion générale

L'objectif des SU est de trouver les meilleurs canaux libres en termes de qualité pour assurer continuellement une très bonne qualité. Cependant, l'objectif du PU est de maximiser son gain c'est à dire qu'il va offrir les canaux libres au plus offrant. La négociation entre les SU et PU se fera dans un premier temps sur deux critères qui est le prix et la qualité.

Dans ce rapport, nous essayons d'apporter une solution à la gestion du spectre en utilisant une technique basée deux approches de résolution exactes. Nous avons choisi la programmation dynamique et B&B comme méthode exacte et on a opté à une métaheuristique qui est l'approche génétique afin d'élaborer notre étude comparative selon différents critères que nous avons fixés (temps de traitement, coût obtenu et qualité).

A noter ici que nous avons opté pour la parallélisation de la programmation dynamique. Les résultats des calculs montrent que les méthodes exactes donnent des solutions de bonne qualité dans un temps de calcul raisonnable, et que l'approche génétique peut générer de solutions moins bonne dans un délai d'exécution raisonnable.

Les objectifs généraux de ce travail de recherche sont de deux ordres. Le premier est d'étudier les relations entre le calcul parallèle et les méthodes exactes afin de mieux comprendre les bénéfices que peuvent apporter le parallélisme dans la résolution de problèmes d'optimisation. La deuxième est d'identifier et de mettre en œuvre des solutions parallèles performantes à un problème d'optimisation combinatoire pratique, c'est-à-dire un problème d'allocation la meilleur bande spectral pour les utilisateurs secondaire des réseaux radio cognitive.

Nous avons résolu le problème à choix multiple en utilisant des méthodes exactes. L'algorithme est basé sur une approche principale à l'utilisation des opérations algébrique de Pareto, éliminant si possible nombre de configuration en entrée, pour ne laisser que les combinaisons non dominées.

Parmi les perspectives à court terme, il nous semble important de porter des améliorations sur notre algorithme de séparation et d'évaluation est de lui combiné aux opérations algébriques de Pareto.

Une autre direction de recherche qui nous semble intéressante c'est de réfléchir sur l'amélioration du B&B, nous pensons au parallélisme de la méthode proposée sur plus de threads. Afin de pouvoir considérer plus de nœuds dans la version améliorée que nous avons présentée et le domaine de recherche des solutions.

Références bibliographiques

- [1] A. Amraoui, W. Baghli, and B. Benmammar, "Amélioration de la fiabilité du lien sans fil pour un terminal radio cognitive mobile".
- [2] A. AMRAOUI "Vers une architecture multi-agents pour la radio cognitive opportuniste".
- [3] Amraoui Asma, Baghli Wassila, "Synthèse de profils applicatifs dans le cadre des réseaux radio cognitive".
- [4] A. Schärli, Décider sur plusieurs critères. Panorama de l'aide à la décision.
- [5] Badr Benmammar, Asma Amraoui, "Réseaux de radio cognitive: Allocation des ressources radio et accès dynamique au spectre".
- [6] B. Ahlam, L. Mohamed, and M. B. Amel, "Vers l'auto-gestion d'un réseau de radio cognitive".
- [7] B. Benmammar, "Présentation de la radio cognitive".
- [8] Bendella Med Saléh, "Gestion de spectre dans les réseaux de radio cognitive par la formation de coalitions".
- [9] B. Benmammar, A. Amraoui, F. Krief, "A Survey on Dynamic Spectrum Access Techniques in Cognitive Radio Networks", International Journal of Communication Networks and Information Security (IJCNIS), August 2013.
- [10] Bellman, R. Dynamic programming princeton university press princeton.
- [11] Barnhart, Cynthia, et al. "Branch-and-price: Column generation for solving huge integer programs." Operations research (1998).
- [12] Czyzak, P., & Jaskiewicz, A. 1998. Pareto simulated annealing a metaheuristic technique for multiple-objective combinatorial optimisation. Journal of Multi-Criteria Decision Analysis.
- [13] Fukasawa, Ricardo et al. "Robust branch-and-cut-and-price for the capacitated vehicle routing problem." Mathematical programming (2006).
- [14] J. Palicot, "De la radio logicielle à la radio intelligente", Collection Technique et Scientifique des télécommunications, Hermes/lavoisier, 2010.

- [15] J. Mitola and G. Maguire, "Cognitive radio: Making software radios more personal", IEEE Personal Communications, August 1999.
- [16] Ibtissem Larbi, Badr Benmammar, "Négociation de spectre dans les réseaux de radio cognitive", Rapport de recherche, Laboratoire de télécommunications de Tlemcen (LTT), Université Abou Bekr Belkaid Tlemcen, 2013.
- [17] I. Ngom et L. Diouf, "La radio cognitive", Université Lille 1 USTL, 2008.
- [18] G. M. Padberg, «A branch-and-cut algorithm for the resolution of largescale symmetric traveling salesman problems,» SIAM review, 1991.
- [19] T. R. Newman, R. Rajbanshi, A. M. Wyglinski, J. B. Evans and G. J. Minden, "Population adaptation for genetic algorithm-based cognitive radios," in IEEE Proc. Cognitive Radio Oriented Wireless Networks and Communications, Orlando, FL, Aug. 2007.
- [20] U. Mir, "Utilization of Cooperative Multiagent Systems for Spectrum Sharing in Cognitive Radio Networks", PHD THESES, September 2011 .
- [21] Y.Collette and P.Siarry, "Optimization multiobjectif", Eyrolles, 2002.
- [22] Lawler, Eugene L, and David E. Wood. "Branch-and-bound methods: A survey."

Abstract : The key idea of spectrum decision in cognitive radio networks is the selection of the best available frequency band to meet the requirements of secondary users, without interfering with the transmission of licensed or primary users. This difficult task requires very good cooperation between users with different demands for the best use of the spectrum channels of different characteristics in a heterogeneous network.

In this manuscript, we studied the multicriteria decision scenario for selecting the best frequency band and we applied two exact methods, dynamic programming and the Branch and Bound method, and an approximate method that is the genetic algorithm, in order to establish a comparative study. Then we used a parallel implementation on the exact methods to improve the execution time. Our simulations approve the desired results, show time-saving processing.

Keywords

Cognitive radio - dynamic programming - multi-criteria decision - Branch and bound

Résumé : L'idée clé de la décision de spectre dans les réseaux radio cognitifs est la sélection de la meilleure bande de fréquences disponible pour répondre aux besoins des utilisateurs secondaires, sans interférer avec la transmission des utilisateurs sous licence ou primaires. Cette tâche difficile nécessite une très bonne coopération entre les utilisateurs ayant des demandes différentes pour la meilleure utilisation des canaux spectraux de différentes caractéristiques dans un réseau hétérogène.

Dans ce manuscrit, nous avons étudié le scénario de décision multicritère pour sélectionner la meilleure bande de fréquence et appliqué deux méthodes exactes, la programmation dynamique et la méthode Branch and Bound, et une méthode approximative qui est l'algorithme génétique pour établir une étude comparative. Ensuite, nous avons utilisé une implémentation parallèle sur les méthodes exactes pour améliorer le temps d'exécution. Nos simulations approuvent les résultats souhaités, montrent un gain de temps traitement.

Mots clés

Radio cognitive - Programmation dynamique - branche et borne - décision mutli critère (topsis).

ملخص

الفكرة الرئيسية لقرار الطيف في الشبكات الراديوية الإدراكية هي اختيار أفضل نطاق ترددي متاح لتلبية متطلبات المستخدمين الثانويين، دون التدخل في إرسال المستعملين المرخصين أو الأوليين حيث تتطلب هذه المهمة الصعبة تعاوناً جيداً للغاية بين المستخدمين ذوي المطالب المختلفة لاستخدام أفضل لقنوات الطيف ذات الخصائص المختلفة في شبكة غير متجانس.

في هذه الاطروحة درسنا سيناريو قرار متعدد المسارات لاختيار أفضل نطاق تردد. الهدف الرئيسي هو السماح للنهج المعرفي باستخدام طريقة دقيقة والحصول على نتائج أفضل مقارنة باستخدام طرق تقريبية

لهذا، قمنا بتطبيق طريقتين دقيقتين، برمجة ديناميكية وأسلوب الفروع والضم، وطريقة تقريبية هي الخوارزمية الجينية، من أجل إنشاء دراسة مقارنة ثم استخدمنا التنفيذ المتوازي على الطرق الدقيقة لتحسين وقت التنفيذ. تعتمد عمليات المحاكاة لدينا على النتائج المرغوبة، وتعرض المعالجة الموفرة للوقت.

الكلمات المفتاحية: الراديو الادراكي – دعم قرارات متعددة المعايير – البرمجة الديناميكية – طريقة الفصل والتقييم