



République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: Réseaux et Systèmes Distribués (R.S.D)

Thème

**Application des méthodes exactes pour
l'allocation des ressources dans la radio
cognitive**

Réalisé par :

- *Kassaa Ahmed*

Présenté le devant le jury composé de .

- M Boudefla (Président)
- M BENMOUNA YOUCEF. (Encadrant)
- M Benazzouz (Examineur)

Année universitaire: 2021-2022

Remerciements

La rédaction de ce mémoire et sa soutenance marquent la fin d'une belle aventure de cinq ans à l'université de Tlemcen.

Louange à Dieu de m'avoir donné la force d'accomplir ce travail.

الحمد لله حمدا كثيرا

A travers ce modeste travail, je tiens à remercier vivement mon encadrant monsieur BENMOUNA Youcef pour son appui ; suivi continu ; et ses précieux conseils durant la réalisation de ce projet.

Je tiens à témoigner toute ma reconnaissance à monsieur (nom et prénom du président) d'avoir accepté de présider ce travail.

*Je souhaite adresser mes sincères remerciements à mon équipe de formation ,
Pour la qualité des enseignements fournis tout au long de
ces deux dernières années.*

*Je voudrais remercier messieurs (examineurs) d'avoir accepté d'évaluer ce
travail.*

Enfin, merci à nos familles et nos proches

 KASSAA AHMED

Dédicace

Je dédie ce mémoire

*À ma grande famille surtout mes très
chers parents*

*À ma chère femme qui m'a soutenu
durant*

ma formation

À mes neveux et mes nièces .

À mes amis.

Table des matières

1	Introduction Générale.....	1
1.1	Description du projet.....	1
1.2	Problématique :	1
	Chapitre I : La radio cognitive	2
1	Introduction	3
2	Radio logicielle	3
3	La radio logicielle restreinte.....	4
4	Radio cognitive	4
4.1	Définition	4
4.2	Historique.....	5
5	Relation entre radio cognitive et radio logicielle restreinte	6
6	Composantes de la radio cognitive.....	7
6.1	Emetteur / Récepteur.....	7
6.2	Analyseur de spectre	7
6.3	Apprentissage et extraction de connaissances.....	8
6.4	Prise de décision.....	8
7	Cycle de cognition et fonctionnement de la radio cognitive.....	8
7.1	Phase d'observation.....	10
7.2	Phase d'orientation.....	10
7.3	Phase de planification.....	10
7.4	Phase de décision	11
7.5	Phase d'action	11
7.6	Phase d'apprentissage	11
8	Conclusion.....	11
	Chapitre II : Les méthodes de résolution.	12
1	Introduction	13
2	Les méthodes de résolution	13
2.1	Les méthodes approchées.....	14
2.1.1	Les heuristiques.....	14
2.1.1.1	Les heuristiques de construction.....	14
2.1.1.2	La méthode de Descente	15
2.1.2	Méta-heuristique.....	16
2.2	Méthodes exactes (appelées aussi complètes).....	16
2.2.1	La méthode de programmation dynamique.....	17
2.2.2	La méthode de génération de colonne.....	17
2.2.3	L'algorithme A*.....	17

2.2.4	La méthode de Branch and Bound	17
2.2.4.1	Le principe d'évaluation	19
2.2.4.2	Le principe de séparation	19
2.2.4.3	La stratégie de parcours	19
3	Conclusion.....	20
	CHAPITRE III : Expérimentation.....	21
1	Introduction	22
2	Environnement de travail	22
2.1	Environnement matériel	22
2.2	Environnement logiciel	22
3	Présentation du problème	22
4	Méthodes de résolution	24
4.1	Les algorithmes génétiques	24
4.2	Fonctionnement de l'algorithme génétique.....	25
4.2.1	La création de la population initiale.....	25
4.2.2	Codage et population initiale.....	25
4.2.3	L'évaluation des individus.....	27
4.2.4	La création de nouveaux individus.....	27
4.2.4.1	La sélection	27
4.2.4.2	Les croisements :.....	28
4.2.4.3	Mutation.....	29
4.2.5	L'insertion des nouveaux individus dans la population.....	29
4.2.6	Réitération du processus.....	30
4.3	La programmation dynamique	30
5	Expérimentation	32
5.1	L'algorithme génétique	32
5.1.1	Ajustements des paramètres	32
5.2	La programmation dynamique :	36
6	Comparaison.....	37
6.1	Scénario des tests	37
6.2	Comparaison des résultats obtenus en termes de gain:	38
6.3	Comparaison des résultats obtenus en termes de temps d'exécution :.....	39
7	Présentation de l'application :	41
8	Conclusion.....	42
	CONCLUSION GENERALE :.....	44

Liste des figures

Figure 1: Evolution de la radio matérielle à la radio logicielle.....	4
Figure 2: Étapes développements dans la radio cognitive.	5
Figure 3: Relation entre radio cognitive et radio logicielle restreinte.....	6
Figure 4: Composantes de la radio cognitive.....	7
Figure 5: Cycle cognitif, FR désigne la fréquence radio.....	8
Figure 6: Cycle cognitif simplifié.	9
Figure 7: Cycle de cognition de Mitola.....	10
Figure 8: Classes des méthodes de résolutions.	13
Figure 9 : L'algorithmme glouton de rendu de monnaie.....	15
Figure 10 : l'algorithmme de séparation et évaluation.....	18
Figure 11 : le croisement des gènes.....	28
Figure 12 : Histogramme solution optimal par la taille de population.....	32
Figure 13 : Fonction meilleur Individu.	33
Figure 14 : Fonction de croisement.....	33
Figure 15 : Fonction mutation.....	34
Figure 16 : Fonctionnement de la méthode génétique.	35
Figure 17 : Fonction de la programmation dynamique.....	36
Figure 18 : Graphe de temps d'exécutions par rapport à le nombre d'utilisateur.....	37
Figure 19 : Histogramme de résultats de la programmation dynamique et l'algorithmme génétique.....	38
Figure 20 : Le graphe de résultats de la programmation dynamique et l'algorithmme génétique.....	39
Figure 21 : Histogramme de temps d'exécution de programmation dynamique et l'algorithmme génétique.....	40
Figure 22 : le graphe de temps d'exécution de la programmation dynamique et l'algorithmme génétique.....	40
Figure 23 : Page d'accueil d'application.....	41
Figure 24 : Interface des résultats.....	42

Liste des tableaux

Tableau 1 : Liste des étapes.....	15
Tableau 2 : le nombre de canal et prix associé aux différents utilisateurs secondaire.	23
Tableau 3 : la première sélection d'utilisateurs secondaires.	26
Tableau 4 : la 2 eme sélection d'utilisateurs secondaires.....	26
Tableau 5 : la 3eme sélection d'utilisateurs secondaires.....	26
Tableau 6 : codage des trois résultats.....	27
Tableau 7 : Sélection les deux meilleurs individus dans la population.....	28
Tableau 8 : Les individus parents.....	29
Tableau 9 : Les individus enfants.....	29
Tableau 10 : le nombre de canal et prix associé aux différents utilisateurs secondaire.	31
Tableau 11 : Une matrice des solutions optimales de tout problème $KP(i,nbr_canaux)$	31
Tableau 12 : le choix de meilleure population	32
Tableau 13 : Les résultats des 10 scénarios de la méthode génétique.....	35
Tableau 14 : Les résultats des 10 scénarios de la programmation dynamique.	36
Tableau 15:les résultats de la programmation dynamique et l'algorithme génétique.	38
Tableau 16 :Le temps d'exécution de la programmation dynamique et l'algorithme génétique	39

1 Introduction Générale

1.1 Description du projet

En ce début de troisième millénaire, le domaine des réseaux de transmission a connu une profonde évolution. Les systèmes de communications, et plus particulièrement les systèmes de communications sans fil, font partie des technologies qui ont révolutionné notre mode de vie.

Au fil des ans, les réseaux sans fil sont devenus de plus en plus précieux avec la propagation rapide de l'utilisation des systèmes de communication sans fil et l'expansion de l'internet mobile à travers le monde.

Ceci pousse les chercheurs à développer et améliorer un nouveau service de radio communication avec une meilleure qualité et en même temps réduire son impact environnemental causé par les risques d'interférences et de la saturation du spectre électromagnétique, ce qui rend le spectre encombré. Ce nouveau service est la radio cognitive (RC).

En 1998 à l'Institut royal de technologie dans un séminaire à KTH, Joseph Mitola III expose une nouvelle approche de communication dans les réseaux sans fil c'est la Radio Cognitive.

La capacité de cette dernière offre la possibilité d'interagir avec son environnement radio afin de s'y adapter, de détecter les fréquences libres et de les exploiter.

« Une radio cognitive peut connaître, percevoir et apprendre de son environnement puis agir pour simplifier la vie de l'utilisateur ». Joseph Mitola.

1.2 Problématique :

Pour éviter les problèmes connus de communication dans les réseaux sans fils, la nouvelle approche RC prend une structure d'allocation des ressources, pour cela la radio cognitive utilise plusieurs méthodes pour améliorer cette structure. Afin d'aboutir à ces objectifs ; ce problème d'optimisation combinatoire peut être considéré comme un problème de sac à dos etc.

En ingénierie, plusieurs problèmes d'optimisation combinatoire sont définis pour la recherche d'une solution optimale parmi un ensemble de solutions candidate pour un problème particulier donné. Ils existent dans la littérature deux grandes méthodes de résolution de ces problèmes, **Exactes** et **Approchées**.

Dans ce travail nous proposons une méthode exacte qui est la programmation dynamique, afin de résoudre ce problème combinatoire et apporter une meilleure solution au problème d'allocation des structures dans les réseaux radio cognitives.

Dans ce PFE, nous tentons d'apporter une meilleure solution à la gestion du spectre, en termes de gain et de temps d'exécution. Pour cela nous avons proposé une méthode exacte pour l'optimisation de l'allocation des ressources,

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

Ce mémoire est structuré comme suit :

- Dans **le premier chapitre** : nous apportons le concept de radio logiciel et de SDR, celle-ci donnant une naissance de radio cognitive
- Dans **le deuxième chapitre** : on détaillera les différentes méthodes de résolution approximatives en plus on apportons le concept de la méthode programmation dynamique .
- **Le troisième chapitre** : est dédié à la présentation des expérimentation et les résultats obtenus, tout cela couronné par une conclusion et des perspectives

Chapitre I :

La radio cognitive

1 Introduction

En télécommunication, une radio intelligente ou radio cognitive CR est une forme de communication sans fil dans laquelle un émetteur/récepteur capable d'adapter certains de ses paramètres automatiquement en fonction de son environnement. Il peut détecter intelligemment les canaux de communication qui sont en cours d'utilisation et ceux qui ne le sont pas, et peut se déplacer dans les canaux inutilisés.

Ceci permet d'optimiser l'utilisation des fréquences radio disponibles (RF) du spectre tout en minimisant les interférences avec d'autres utilisateurs. Cette capacité permet d'adapter chaque appareil aux conditions spectrales du moment et offre donc aux utilisateurs un accès plus souple, efficace et complet à cette ressource. Cette approche peut améliorer considérablement le débit des données et la portée des liaisons sans augmenter la bande passante ni la puissance de transmissions.

La radio cognitive offre également une solution équilibrée au problème de l'encombrement du spectre en accordant d'abord l'usage prioritaire au propriétaire du spectre, puis en permettant à d'autres de se servir des portions inutilisées du spectre. [1]

Ce chapitre porte sur une présentation détaillée de la radio cognitive, où nous découvrons sa relation avec la radio logicielle.

2 Radio logicielle

Depuis quelques années, la multiplication des réseaux sans-fil en nombre et en type a fortement fait évoluer les technologies et les méthodes utilisées pour leur développement. C'est ainsi que la radio logicielle est apparue. [2]

La radio logicielle (ou software defined radio) a été conceptualisée à partir du début des années 1990 par Joseph Mitola, le SDR est différente de la radio matérielle classique. En effet la radio matérielle impose de choisir le composant en fonction de ce qu'il est souhaité faire avec le récepteur ou l'émetteur. Par exemple, si l'on souhaite communiquer par Bluetooth, notre choix va se tourner vers un module Bluetooth, et on ne pourra donc recevoir que du Bluetooth.[3]

La radio logicielle a changé tout cela. L'idée est de réaliser tous les traitements par un programme informatique c'est à dire les fonctions typiques de l'interface radio généralement réalisées en matériel, telles que la fréquence porteuse, la largeur de bande du signal, la modulation et l'accès au réseau sont réalisés sous forme logicielle. La radio logicielle moderne intègre également l'implantation logicielle des procédés de cryptographie, codage, correcteur d'erreur, codage source de la voix, de la vidéo ou des données, donc, les interfaces radio peuvent, en principe, être adaptées aux besoins d'un service particulier pour un usager particulier dans un environnement donné à un instant donné. [1]

L'objectif principale de la radio logicielle est d'intégrer toute les fonctionnalités en logiciel, en imposant des phases intermédiaires combinant les anciennes et les nouvelles techniques, on parle alors de radio logicielle restreinte.[1]

Les contraintes de puissance de calcul, de consommation électrique, de coûts, etc. imposent actuellement de passer par cette phase intermédiaire.

3 La radio logicielle restreinte

La radio logicielle restreinte qu'on nomme également "Software Defined Radio" ou SDR est un système radio communication qui peut modifier dynamiquement certaines de ses caractéristiques de façon logiciel [4]. Il peut aussi modifier sa fréquence porteuse qui va lui permettre de transmettre et recevoir des signaux sur une large bande de fréquence, en plus le type de modulation et le niveau de puissance peuvent être modifier avec le même matériel.

La radio logicielle restreinte est un système permettra alors de s'adapter à n'importe quelle bande de fréquence et recevoir n'importe quelle modulation en utilisant le même matériel.

Les opportunités qu'offre le SDR lui permettent de résoudre des problèmes de la gestion dynamique du spectre. Les équipements SDR peuvent fonctionner dans des réseaux sans fil hétérogènes c'est-à-dire qu'un SDR idéal peut s'adapter automatiquement aux nouvelles fréquences et aux nouvelles modulations [5].

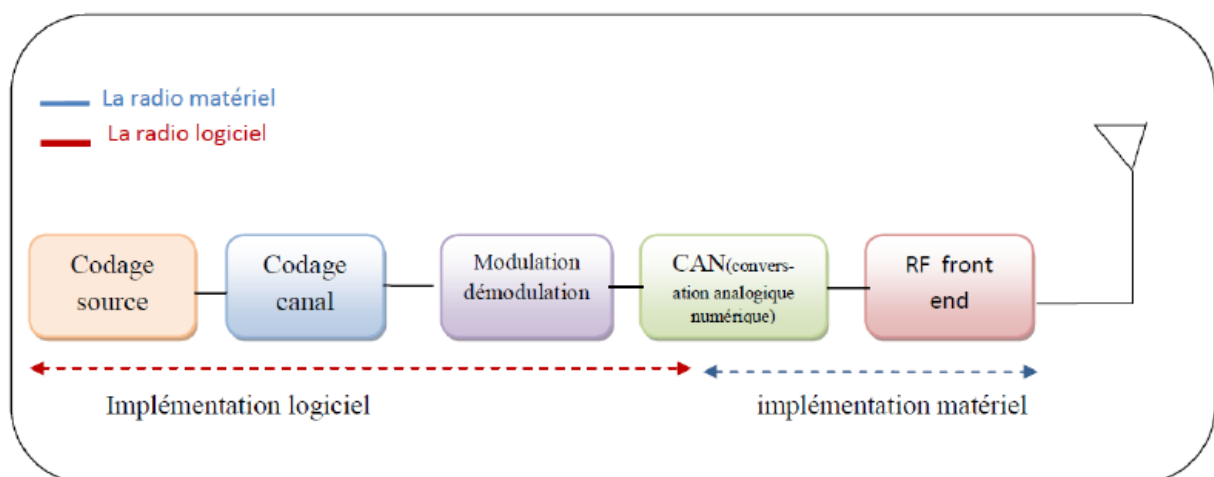


Figure 1: Evolution de la radio matérielle à la radio logicielle.

4 Radio cognitive

4.1 Définition

Commençant par la définition précise de la cognition, qui est une combinaison de nombreux processus mentaux qui se liés à la fonction de la connaissance et comprend la mémoire, le langage, le raisonnement, l'apprentissage, l'intelligence, la résolution de problèmes, la prise de décision, la perception ou l'attention, ces processus allant de l'analyse perceptive de l'environnement à la commande motrice.

La radio cognitive est un type spécial d'émetteur-récepteur radio intelligent qui détecte automatiquement tous les différents canaux radio disponibles sur le spectre, elle facilite les modifications des paramètres de réception ou de transmission, en permettant la transmission simultanée de plusieurs communications radio supplémentaires dans une plage donnée d'un site. Ce processus est connu sous le nom de **gestion dynamique du spectre**.

Le but principal de la radio cognitive est de détecter et de partager le spectre inutilisé avec d'autres systèmes sans créer d'interférences nuisibles. Cela comprend la recherche du meilleur spectre disponible pour répondre aux besoins de communication des utilisateurs. [6]

4.2 Historique

Le concept de la radio cognitive a été présenté officiellement par Joseph Mitola à un séminaire à KTH, l'Institut royal de technologie, en 1998, publié plus tard dans un article de Mitola et Gerald Q. Maguire, Jr en 1999 [5] J. Mitola and G. Q. Maguire, « Cognitive radio : making software radios more personal, IEEE Personal Communications ». puis vient le IEEE en 2008 [4] qui ajoute à cette définition l'aspect décisionnel grâce à des données primaires et un objectif prédéfini.

Mitola combine son expérience de la radio logicielle ainsi que sa passion pour l'apprentissage automatique et l'intelligence artificielle pour mettre en place la technologie de la radio cognitive. D'après lui : Une radio cognitive peut connaître, percevoir et apprendre de son environnement puis agir dans le but de simplifier la vie de l'utilisateur [1].

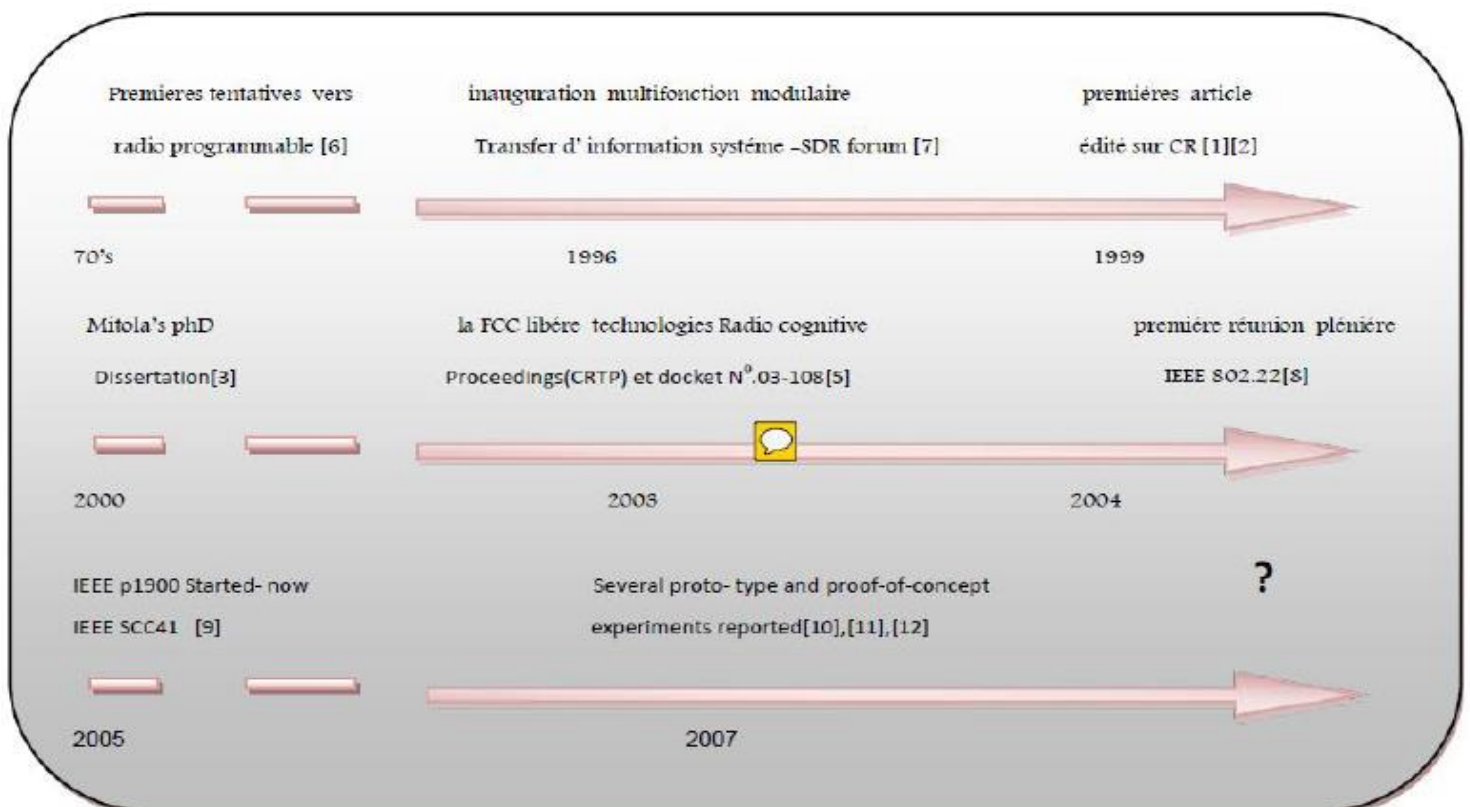


Figure 2: Étapes développements dans la radio cognitive.

5 Relation entre radio cognitive et radio logicielle restreinte

La radio cognitive traite un ensemble des caractéristiques telles que la capacité d'adaptation (bande passante, capacité, modulation et fréquence porteuse) en fonction de l'environnement radio, des besoins de l'utilisateur, de l'état du réseau, La situation et La géolocalisation etc

Software Defined Radio (RL) associée à une flexibilité logicielle capable d'introduire le changement de contexte (changement de service, de traitement) et introduit les notions de reconfiguration dit dynamique, inclut la reconfiguration sans interruption de service, et l'aspect d'adaptabilité ou de modularité.

La radio logicielle est en mesure d'offrir les avantages de flexibilité, reconfigurabilité et de portabilité inhérente à l'aspect d'adaptation de la radio cognitive. Par conséquent, cette dernière doit être mis en œuvre autour d'une radio logicielle. En d'autres termes, la radio logicielle est une "technologie habilitante" pour la radio cognitive. ([1])

La relation entre RC et SDR est déterminée par un modèle simple illustré à la figure 3. Ce modèle représente les éléments de la RC et entoure le support SDR et le cognitif engin qui est nécessaire à la prise de décision et à l'apprentissage de l'environnement radio qui utilise efficacement les ressources disponibles.

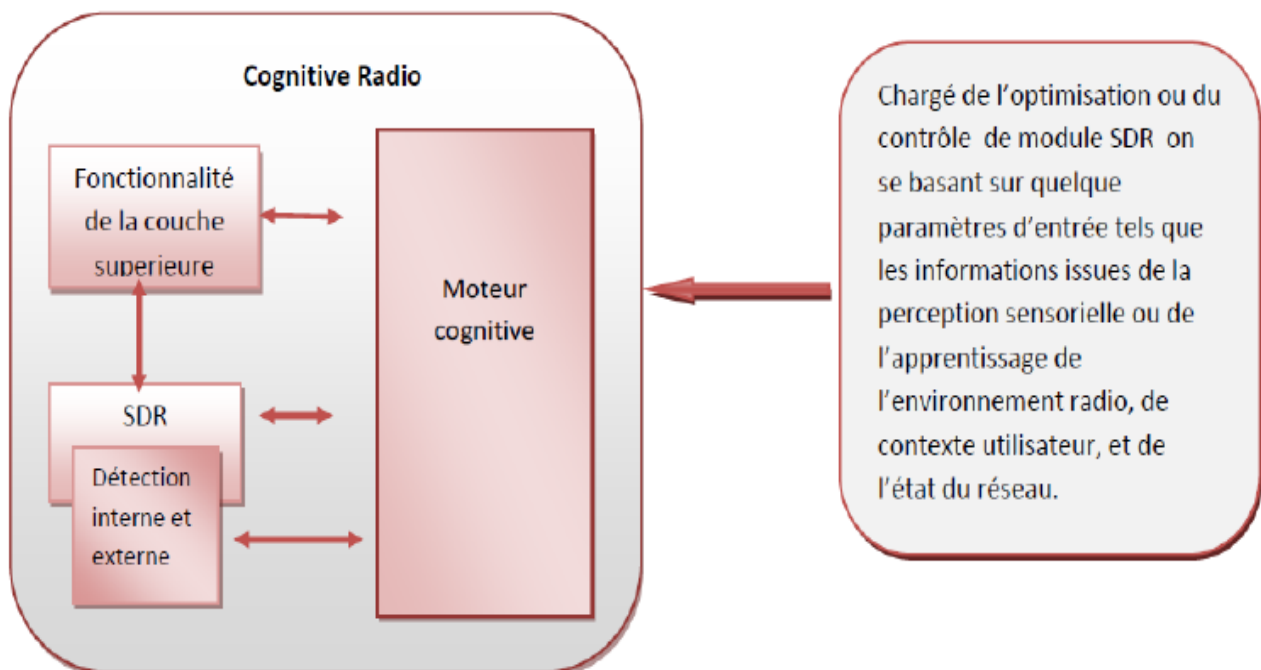


Figure 3: Relation entre radio cognitive et radio logicielle restreinte.

6 Composantes de la radio cognitive

La radio cognitive assure son bon fonctionnement grâce à des composants spécifiques, ces composants sont présentés en points ci-dessous :

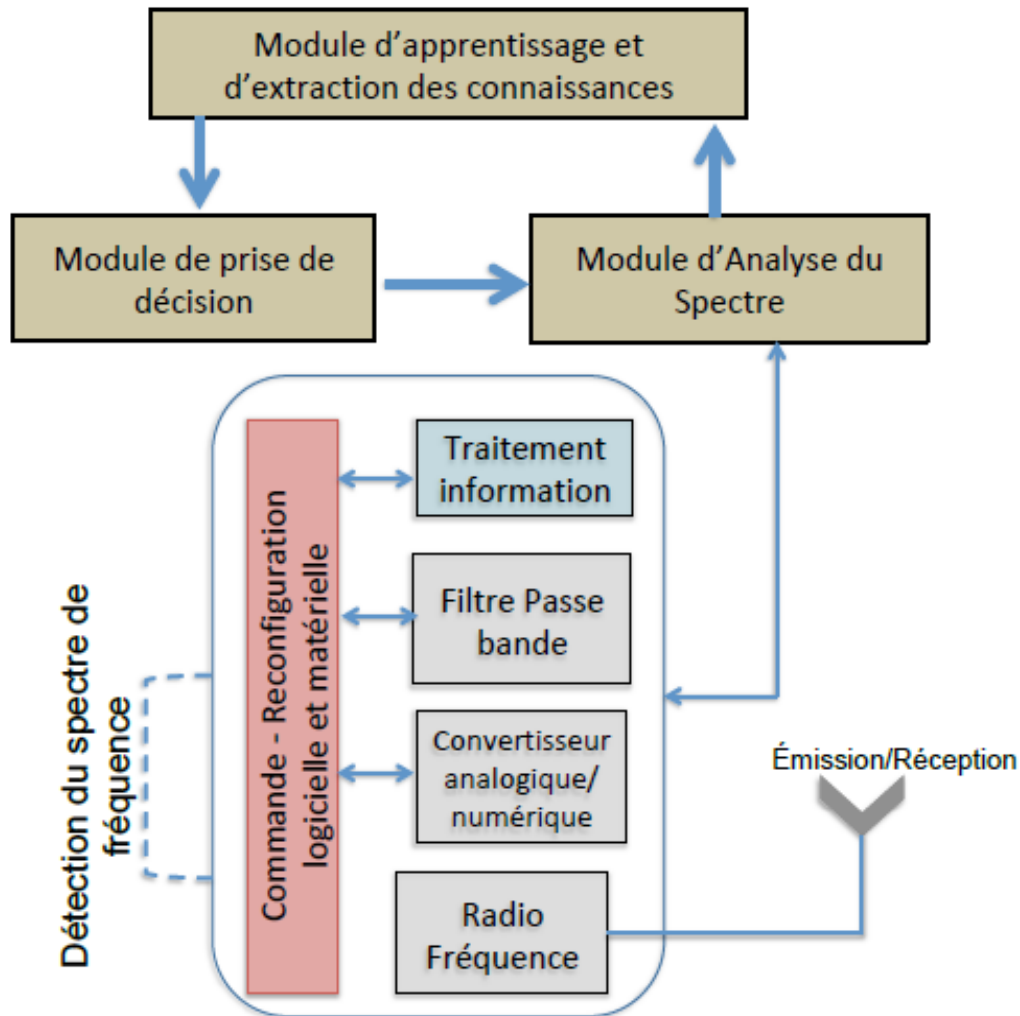


Figure 4: Composantes de la radio cognitive.

6.1 Émetteur / Récepteur

Le rôle de ces modules est de détecter intelligemment les canaux de communications libres et occupés, il peut aussi se déplacer dans les canaux libres, ce qui permet une meilleure gestion du spectre radio cognitive.

6.2 Analyseur de spectre

L'analyseur de spectre (Spectrum analyser) utilise les signaux mesurés pour analyser l'utilisation du spectre pour arriver à détecter la signature du signal qui provient d'un utilisateur licencié sur le canal, appelé aussi utilisateur primaire (PU), et trouver les espaces blancs du spectre (les espaces libres ou période d'inactivité sur le spectre licencié appelé aussi les canaux non utilisés par PU) pour être alloués par les utilisateurs secondaires (SU) afin d'assurer une transmission sans interférence ni dégradation.

6.3 Apprentissage et extraction de connaissances

L'apprentissage et l'extraction de connaissances utilisent les algorithmes d'apprentissage pour construire une base de connaissance de l'environnement ambiant basée sur les informations collectées par l'analyseur du spectre, qui est ensuite utilisée pour configurer les paramètres de transmission pour atteindre un objectif donné sous diverses circonstances.

6.4 Prise de décision

Une décision optimale est faite grâce à des informations fournies par la base des connaissances de l'environnement qui est construite et entretenue par l'extraction de connaissances et l'apprentissage automatique.

Différentes techniques peuvent être utilisées pour obtenir une décision optimale qui dépend du milieu ambiant, du comportement (coopératif / compétitif) des SUs.

7 Cycle de cognition et fonctionnement de la radio cognitive

Le transpondeur radio cognitive sentir son environnement radio, analyser les informations capturées pour déterminer quelle action est la meilleure en termes de spectre à utiliser et de meilleure stratégie de transmission à adopter. Cela permet à la radio cognitive de surveiller en permanence le changement dynamique de son environnement afin de programmer et de configurer dynamiquement de sorte que les meilleurs canaux sans fil à proximité soient utilisés pour éviter les interférences et la congestion des utilisateurs.

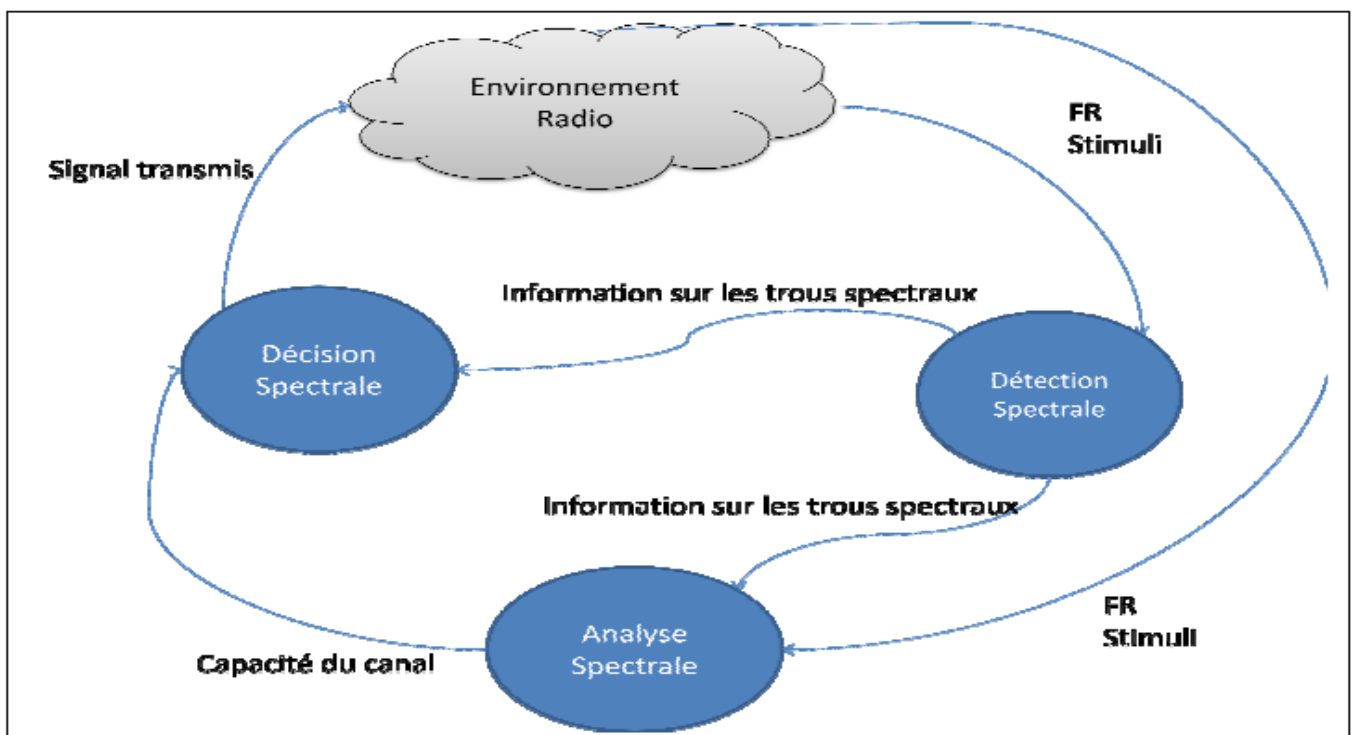


Figure 5: Cycle cognitif, FR désigne la fréquence radio.

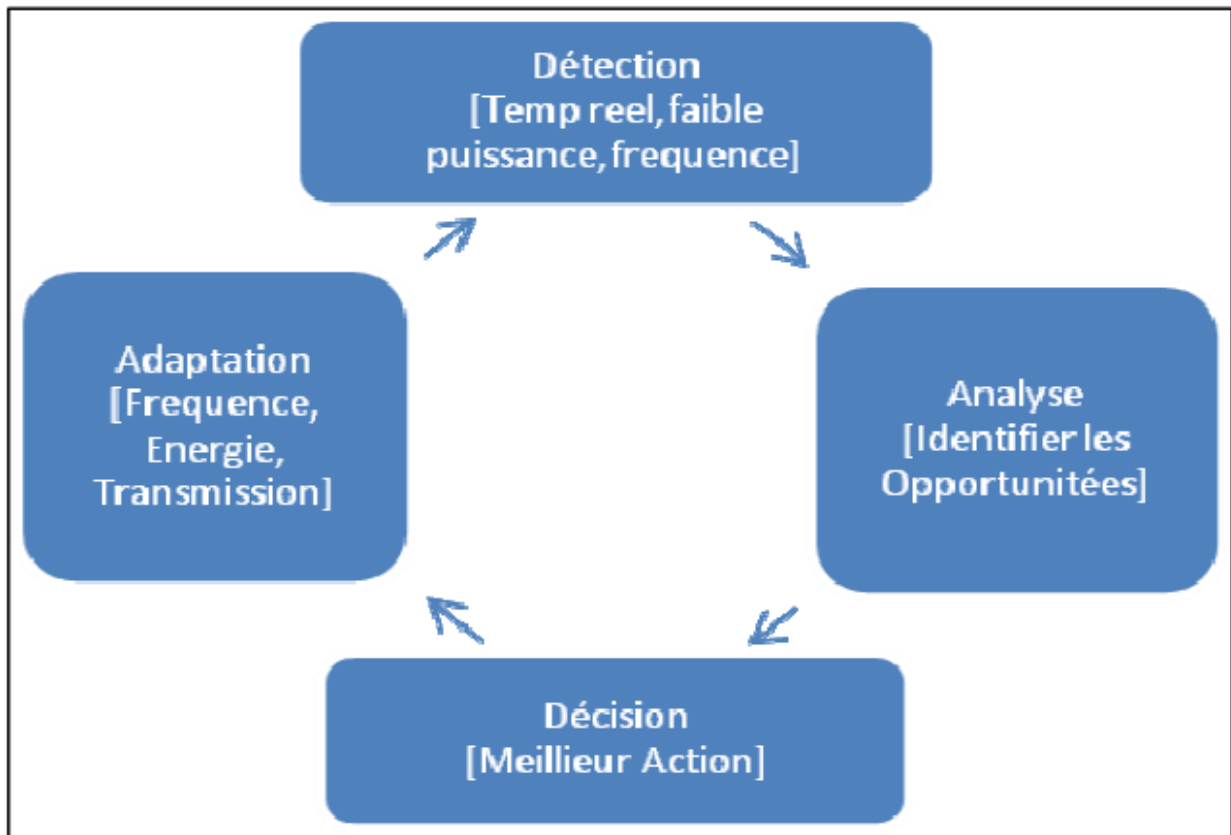


Figure 6: Cycle cognitif simplifié.

Chaque interface radio est supposé être conscient de son environnement. La Figure I.7 c'est un schéma récapitulatif du cycle de concept de la radio idéale qui est proposé par Mitola. La RC analyse son environnement, observe le comportement des différents systèmes communicants sur le réseau, reconnaît les interactions, apprend le but des autres usagers, propose de nouvelles solutions alternatives et agit en conséquence. Elle négocie de nouveaux schémas de communication ainsi que de nouveaux protocoles avec les autres radios opérant sur le réseau.[8]

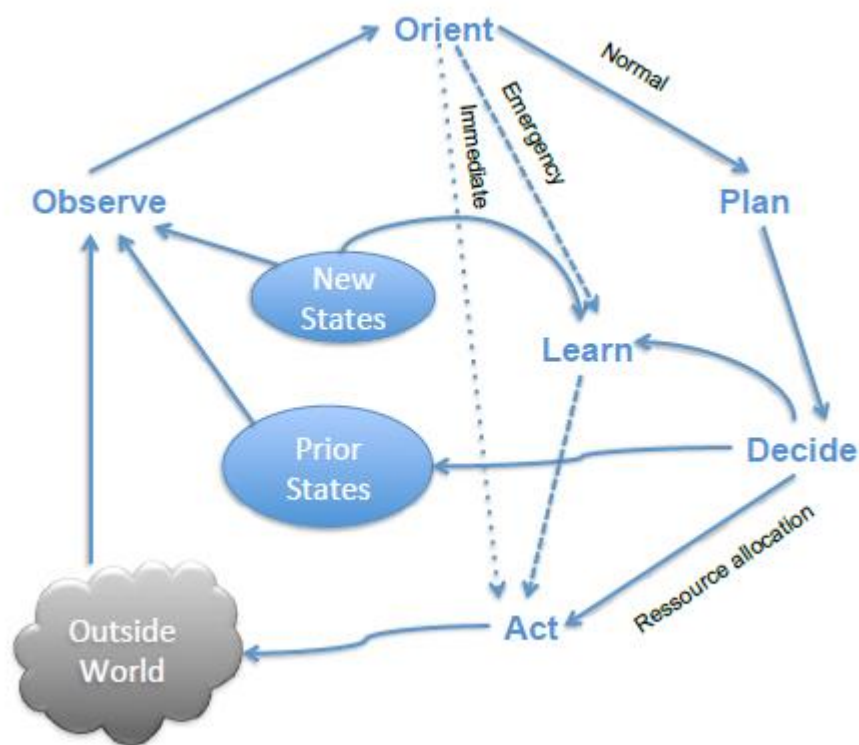


Figure 7: Cycle de cognition de Mitola.

7.1 Phase d'observation

C'est la phase la plus critique de la radio cognitive vu qu'elle offre une observation sur l'état de l'environnement où la radio cognitive, elle mesure l'activité électromagnétique due aux transmissions courantes dans les différentes bandes de fréquences, ensuite capture les paramètres relatifs à telle fréquence, le niveau cumulatif d'énergie et l'activité des utilisateurs. En plus de ces paramètres, la RC détecte l'emplacement, la température, le niveau de lumière des capteurs. Toutes ces informations sont capturées et rassemblées afin d'aboutir à une idée générale et précise sur l'état de l'environnement.

7.2 Phase d'orientation

La phase d'orientation détermine l'importance d'une observation en liant à celle-ci une série connue de stimuli. Cette phase traite les résultats inhérents à la phase d'observation qui sont analogues à la mémoire à court terme (STM), les expériences stockées se fait par reconnaissance des stimuli ou par reliure.

7.3 Phase de planification

La plupart des informations réseau collectées sont traitées avec délibérative plutôt qu'avec réactivité. Un message entrant au réseau serait normalement traité par la génération d'un plan. Le plan devrait également inclure la phase de raisonnement dans le temps. Généralement, les réponses réactives sont préprogrammées ou apprises, tandis que d'autres réactions de délibération sont prévues [1].

7.4 Phase de décision

La phase de décision sélectionne un plan d'allocation parmi un ensemble les plans faisables. La radio peut avertir l'utilisateur d'un message entrant ou reporter l'interruption à plus tard.

7.5 Phase d'action

Cette phase lance les processus sélectionnés qui utilisent les effecteurs sélectionnés qui accèdent au monde extérieur ou aux états internes de la radio cognitive.

L'accès au monde extérieur consiste principalement à composer des messages qui doivent être envoyés dans l'environnement en audio ou exprimés dans différents langages appropriés [1].

7.6 Phase d'apprentissage

L'apprentissage dépend des résultats collectés lors des phases d'observation, et de décisions. L'apprentissage initial, est réalisé à travers la phase d'observation dans laquelle toutes les perceptions sensorielles sont continuellement comparées à l'ensemble de l'expérience antérieure pour continuellement compter les événements et se souvenir du temps écoulé depuis le dernier événement.

La phase d'apprentissage se déclenche quand un nouveau modèle d'allocation est créé en réponse à une action. Par exemple, les états internes antérieurs et courants peuvent être comparés avec les attentes pour en apprendre davantage sur l'efficacité d'un mode de communication.

8 Conclusion

Nous avons présenté dans ce chapitre les notions fondamentales concernant la radio cognitive, c'est un domaine qui combine la télécommunications et l'intelligence artificielle. La radio cognitive offre une observation sur l'état d'environnement, ensuite il analyse les informations capturées pour choisir la meilleure action en termes de spectre à utiliser et la meilleure stratégie de transmission à adopter afin d'atteindre l'objectif désiré sous diverses contraintes.

Dans le chapitre suivant nous abordons les différentes méthodes de résolution, et nous nous intéressons aux méthodes exactes.

Chapitre II :

Les méthodes de résolution.

1 Introduction

La radio cognitive est une forme de communication sans fil permettant de trouver la meilleure solution parmi un ensemble des solutions réalisables afin d'atteindre l'objectif désiré sous diverses contraintes.

La recherche d'une solution optimale dans un ensemble fini est un problème facile en théorie, il suffit simplement d'essayer toutes les solutions ensuite, choisir la meilleure solution en fonction de leurs qualités. Cependant, en pratique, cela peut prendre beaucoup de temps pour lister toutes les solutions possibles, car le temps nécessaire pour trouver la solution optimale (le temps de recherche) peut être très important, par conséquent, les chercheurs ont développé plusieurs méthodes de résolution en recherche opérationnelle et en (IA) .

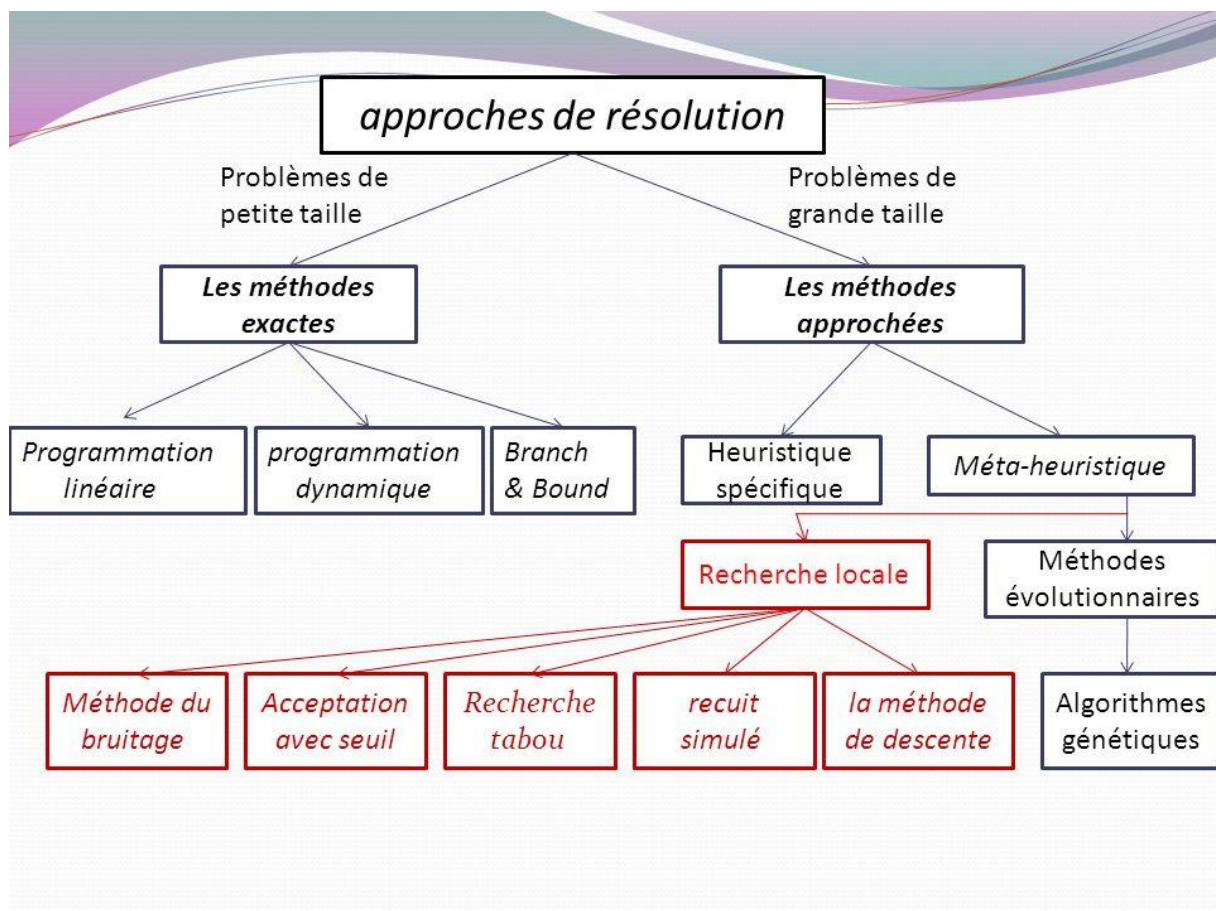


Figure 8:Classes des méthodes de résolutions.

2 Les méthodes de résolution

Le problème du voyageur de commerce, l'ordonnancement de tâches, le problème de la coloration de graphes, le problème de sac à dos etc. ; sont des problèmes modélisés sous la forme d'un problème d'optimisation combinatoire (POC).

En ingénierie, les problèmes d'optimisation combinatoire sont définis comme la recherche d'une solution optimale parmi un ensemble de solutions candidate pour un problème particulier donné.

Les méthodes de résolution se divisent en deux grands groupes de méthodes, les méthodes exactes et les méthodes approchées (figure 8).

2.1 Les méthodes approchées

Les méthodes approchées permettent de trouver de manière rapide une solution (solution réalisable, une valeur approchée de la solution optimale) mais ne garantissent pas que la solution qu'elles fournissent soit optimale pour le problème donné.

Les méthodes approchées est utilisé pour résoudre des problèmes souvent très difficiles à résoudre par des méthodes exactes ; son but de trouve une valeur approchée de celle d'une méthode exacte mais dans un temp plus rapide On distingue deux grandes classes de méthodes : les heuristiques et les méta-heuristiques.

2.1.1 Les heuristiques

Les heuristiques sont des techniques qui sont utilisées pour améliorer l'efficacité d'un processus de recherche qui tente de trouver des solutions à des problèmes complexes. Ces algorithmes parcourt une seule branche de l'arbre de recherche et ne fait pas de retour arrière [9,10]. Une heuristique est spécifique à un problème donné et ne peut pas être généralisée.

Deux types d'heuristiques sont utilisées : les heuristiques de construction (comme les méthodes gloutonnes), qui construit une solution de manière récursive, et les heuristiques de descente, qui à partir d'une solution donnée cherchent un optimum local.[11]

2.1.1.1 Les heuristiques de construction

Les heuristiques de construction ou constructives commencent par une solution construite à partir d'une combinaison vide, et par l'ajout itératif des composants à la sous-solution courante (solution partielle avec une itération donnée).

Les méthodes constructives sont largement utilisées généralement pour calculer des solutions initiales, ou à l'intérieur d'autres procédures par exemple l'algorithme glouton (greedy en anglais) sont les méthodes de construction les plus connues. Elle consiste à effectuer des choix "locaux" qui permettent de réduire le problème à un problème plus simple.

Exemple : On cherche à rendre la monnaie avec le moins de pièces et de billets possible[12], Etant donné le système monétaire suivant.

Pièces:0,01 ; 0,02 ; 0,05 ; 0,1 ; 0,2 ; 1 ; 2.

Billets : 5 ; 10 ; 20 ; 50 ; 100 ; 200 ; 500.

On rendre la monnaie en commençant par la pièce ou le billet avec la plus grande valeur possible (en restant inférieur à la somme à rendre) et on recommence la même opération jusqu'à obtenir une valeur nulle.

On note :

Système : liste des pièces et des billets

Somme : montant à obtenir

Somme_restante : montant qui reste à rendre

Monnaie : liste qui contient les valeurs rendues

L'algorithme glouton de rendu de monnaie peut être décrit comme suit :

1. initialiser monnaie à une liste vide
2. initialiser la somme_restante à somme
3. tant que somme_restante > 0 :
 - On choisit la plus grande valeur dans système inférieure à somme_restante
 - on ajoute cette valeur à monnaie
 - on retire cette valeur à somme_restante
4. renvoyer monnaie

Figure 9 : L'algorithme glouton de rendu de monnaie.

Voici les résultats présentés dans le tableau ci-dessous :

Étapes	liste des monnaies rendues	somme restant à rendre
Initialisation	monnaie=[]	somme_restante=\$53\$
Etape 1	monnaie = \${50}\$	somme_restante=\$3\$
Etape 2	monnaie = \${50,2}\$	somme_restante=\$1\$
Etape 2	monnaie = \${50,2,1}\$	somme_restante=\$0\$

Tableau 1 : Liste des étapes.

2.1.1.2 La méthode de Descente

Les méthodes de descente sont des méthodes très utilisées en IA, ce sont des méthodes de recherche locale les plus simple, qui à partir d'une solution donnée cherchent un optimum local (La meilleur solution possible) dans un délai de temps raisonnable.

Les méthodes de descente sont assez anciennes et leur succès provient de leur rapidité et leur simplicité. A chaque pas de la recherche, cette méthode progresse vers une solution voisine de meilleure qualité. La descente s'arrête quand tous les voisins candidats sont moins bons que la solution courante.

Le défaut majeur de cette méthode est qu'elle s'arrête et fournit une solution avec le premier minimum local rencontré.

Exemple : Pour un problème de minimisation d'une fonction f , la méthode descente peut être décrite comme suit :

1. Solution initiale s ;
2. Repeter :
3. Choisir s' dans un voisinage $N(s)$ de s ;
4. Si $f(s') < f(s)$ alors $s := s'$;
5. jusqu'à ce que $f(s') \geq f(s), \forall s' \in N(s)$.

Figure 10 : Algorithme de La méthode de descente.

Pour éviter le blocage au premier minimum local rencontré, l'algorithme peut être modifié par d'autre condition d'arrêt.

2.1.2 Méta-heuristique

Le mot Méta-Heuristique est dérivé de la composition de deux mots grecs: heuristique qui vient du verbe heuriskein et qui signifie 'trouver'. Méta qui est un suffixe signifiant 'au-delà', 'dans un niveau supérieur'.

Des méthodes heuristiques plus poussées ont été développées et ont données naissance à une nouvelle famille d'algorithmes : les méta-heuristiques, elles sont des méthodes inspirées de la nature. Au contraire aux heuristiques, elles s'appliquent à de nombreux problèmes de nature différente.

Le but d'une méta-heuristique est réussir à résoudre les problèmes de grande taille et complexes, permettent généralement de trouver un optimum global pour des problèmes générale. Pour cela, l'idée est à la fois de parcourir l'espace de recherche, et d'explorer les zones qui paraissent prometteuses ; mais sans être « piégé » par un optimum local.[11]

Les métaheuristiques se divisent en deux catégories :

1. Métaheuristiques à solution unique : Ces méthodes traitent une seule solution à la fois, afin de trouver la solution optimale.
2. Métaheuristiques à population de solutions : Ces méthodes utilisent un ensemble de solutions à chaque itération jusqu'à l'obtention de la solution globale.[9]

2.2 Méthodes exactes (appelées aussi complètes)

Les algorithmes exacts sont des méthodes qui permettent de répondre de manière exacte à un problème donnée. Le principe des méthodes exactes se reposent généralement sur la recherche arborescente et sur l'énumération partielle de l'espace de solutions souvent de manière implicite, afin de trouver la solution optimale.

Parmi les avantages de la méthode exacte c'est la certitude de trouver la solution optimale et l'inconvénient de cette méthode c'est le temps d'exécution prohibitif en plus Le nombre de combinaisons augmente avec l'augmentation de la dimension du problème.

Les algorithmes exacts les plus connus sont la programmation dynamique, l'algorithme A^* , les algorithmes de séparation et évaluation (Branch and Bound, Branch and Cut, Branch and Price et Branch and Cut and Price), les algorithmes de retour arrière et les méthode de génération des colonnes.

Nous présentons d'abord dans notre PFE quelques méthodes de la classe des algorithmes Complets ou exacts : programmation dynamique, la génération de colonnes, l'algorithme A* et Branch and Bound.

2.2.1 La méthode de programmation dynamique

Le concept de la méthode de programmation dynamique a été introduit au début des années 1950 par Richard Bellman, elle s'appuie sur un principe simple, appelé le principe d'optimalité de Bellman, la solution optimale de quelques problèmes dépend des solutions optimales à leurs sous-problèmes.

La programmation dynamique c'est une technique générale qui consiste à résoudre un problème en le divisant en sous-problèmes ensuite résoudre ces sous-problèmes de plus petits aux plus grands (calculées de manière ascendante) et stocker la solution optimale de chaque sous-problème afin de combiner leurs solutions et trouver la solution globale.

La programmation dynamique a d'emblée connu un grand succès car elle résout les problèmes les plus répandus comme le problème de sac à dos, la tour de Hanoi, le chemin le plus court par Dijkstra.

2.2.2 La méthode de génération de colonne

En informatique théorique et en recherche opérationnelle, la génération de colonnes est des méthodes utilisées pour des problèmes linéaires de grande taille, la génération de colonnes a été initialement proposée par Ford et Fulkerson en 1958 [10].

L'objectif de cette méthode consiste à décomposer l'ensemble des contraintes en deux sous-ensembles et n'est pas nécessaire d'énumérer toutes les possibilités.

2.2.3 L'algorithme A*

En informatique, plus précisément en intelligence artificielle l'algorithme A* (A star algorithme, qui se prononce A étoile, ou A star en anglais) est un algorithme de planification de chemin qui permet de trouver très rapidement le chemin le plus court entre deux points. L'algorithme A* a été proposé pour la première fois par Peter E. Hart, Nils John Nilsson et Bertram Raphael en 1968, l'algorithme A* a été créé avec l'intention que la première solution trouvée soit la meilleure.[14]

2.2.4 La méthode de Branch and Bound

La technique de séparation et évaluation (plus connues par son nom en anglais Branch and Bound) proposée pour la première fois par Land et Doig en 1960, elle est reprise et adaptée à la plupart des problèmes d'optimisation exacts ou non.[13]

Branch: Branchement (Séparation), séparer un ensemble de solutions en sous-ensembles ;

Bound: Borne (calcul de la borne ou évaluation du nœud), évaluer les solutions d'un sous-ensemble en trouvant une borne supérieure en cas de maximisation et une borne inférieure en cas de minimisation.

PSE (Procédure par Séparation-Evaluation) ou B&B (Branch-and-Bound).

Les méthodes de Branch-and-Bound sont des méthodes basées sur un mécanisme de recherche très intelligent c'est le comportement "Diviser pour régner", l'idée est de partitionner le problème en plusieurs sous-problèmes (ou diviser l'espace des solutions) de façon à ce que la résolution de tous ces sous-problèmes garantisse la résolution du problème initiale.

L'algorithme de Branch and Bound est une méthode arborescente de résolution est souvent utilisé à la place de la programmation dynamique, elle repose sur le parcours arborescent des solutions possibles ensuite en évaluant chaque sous-ensemble de solutions, en représentant les états solutions par un arbre d'états, avec des nœuds, et des feuilles où chaque nœud correspond à une solution partielle, les solutions partielles se forment progressivement en s'enfonçant dans l'arbre et chacune des solutions partielles potentielles possède une borne supérieure et une borne inférieure, Si l'évaluation partielle du nœud x montre que sa qualité est supérieure à la limite supérieure, alors le sous-arbre respectif sera élagué ; Sinon, le nœud sera divisé en nœuds enfants. Ce processus est répété jusqu'à ce que la solution optimale soit trouvée si elle existe.

Une borne supérieure et une borne inférieure de la solution partielle servent à couper quelques branches de l'arbre afin d'éviter d'énumérer de larges classes de mauvaises solutions (d'éviter de traverser tout l'arbre) et en fait cela permet de trouver de bonne solution en un temps de recherche raisonnable.

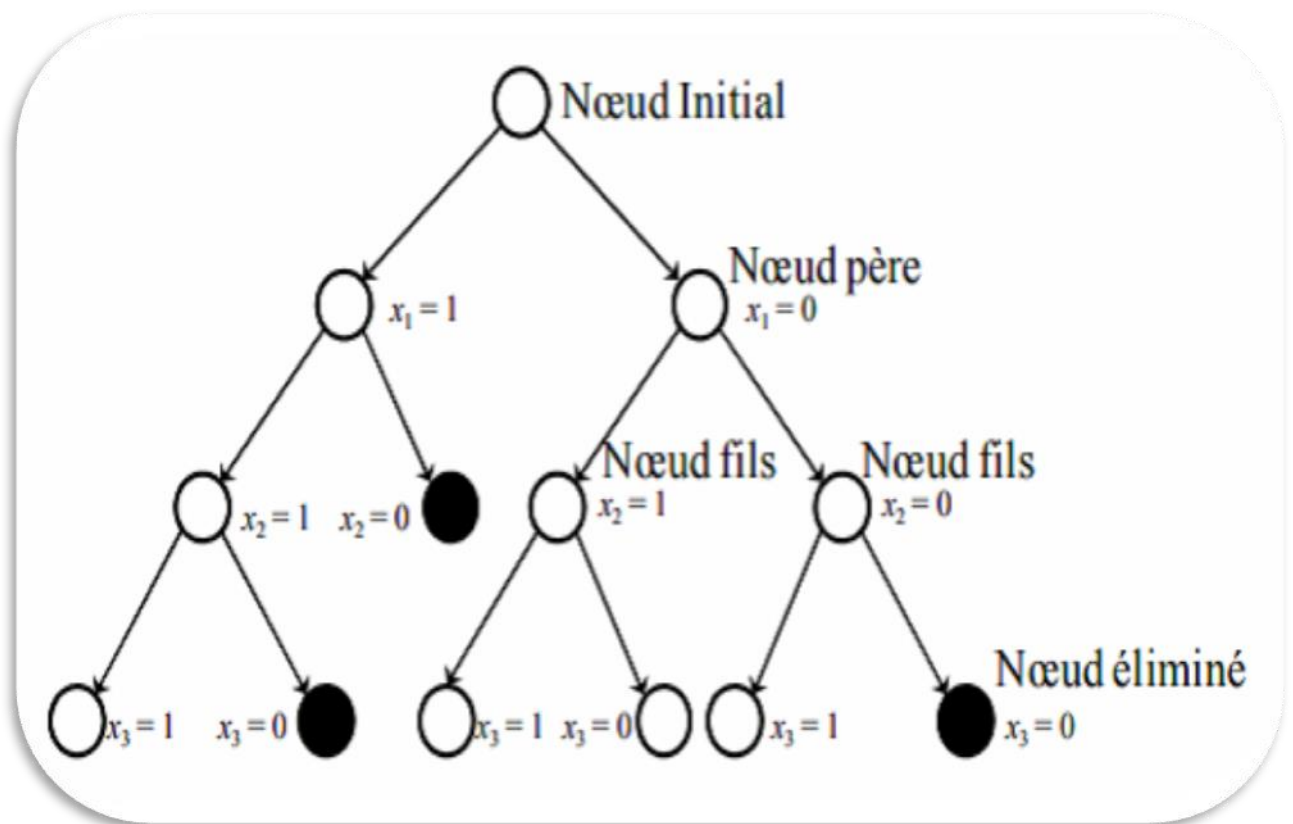


Figure 10 : l'algorithme de séparation et évaluation.

La méthode commence par considérer le problème de départ avec son ensemble de solutions, appelé la racine. Des procédures de bornes inférieures et supérieures sont appliquées à la racine. Si ces deux bornes sont égales, alors une solution optimale est trouvée, et on arrête là. Sinon, l'ensemble des solutions est divisé en deux ou plusieurs sous-problèmes, devenant ainsi des enfants de la racine. La méthode est ensuite appliquée récursivement à ces sous-problèmes, engendrant ainsi une arborescence. Si une solution optimale est trouvée pour un sous-problème, elle est réalisable, mais pas nécessairement optimale, pour le problème départ. Comme elle est réalisable, elle peut être utilisée pour éliminer toute sa descendance : si la borne inférieure d'un nœud dépasse la valeur d'une solution déjà connue, alors on peut affirmer que la solution optimale globale ne peut être contenue dans le sous-ensemble de solution représenté par ce nœud. La recherche continue jusqu'à ce que tous les nœuds soient soit explorés ou éliminés.[15]

L'algorithme Branch and Bound est basé sur trois axes principaux :

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

2.2.4.1 Le principe d'évaluation

Dans l'évaluation, on élimine certains sous-ensembles qui n'ont pas la solution optimale afin de réduire l'espace de recherche. Le but d'une fonction d'évaluation est d'essayer de connaître l'intérêt de l'exploration d'un sous-ensemble de l'arborescence. Le Branch-and-Bound utilise une élimination de branches dans l'arborescence de recherche de la manière suivante :

La recherche d'une solution de coût minimal, consiste à mémoriser la solution de plus bas coût rencontré pendant l'exploration, et à comparer le coût de chaque nœud parcouru à celui de la meilleure solution. Si le coût du nœud considéré est supérieur au meilleur coût, on arrête l'exploration de la branche et toutes les solutions de cette branche seront nécessairement de coût plus élevé que la meilleure solution déjà trouvée.[16]

2.2.4.2 Le principe de séparation

La séparation consiste à dissocier le problème en sous-problèmes selon la technique du « Diviser pour régner »; Le problème est représenté sous la forme d'un arbre qui permet d'énumérer toutes les solutions. Ainsi, nous sommes sûrs de résoudre le problème initial en résolvant tous les sous-problèmes et en mémoriser la meilleure solution trouvée.

La procédure de séparation d'un ensemble s'arrête lorsqu'une des critères est vérifiée :[10]

- 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.

2.2.4.3 La stratégie de parcours

Il existe trois parcours possibles de l'arbre:

- La largeur d'abord :

Cette stratégie permet de privilégier les sommets les plus proches de la racine (avec le moins de profondeur) obtenus avec le moins de séparations du problème initial ; Cette méthode est moins efficace que les deux autres stratégies.

-Le meilleur d'abord :

Cette stratégie préfère explorer des sous-problèmes avec la meilleure borne. Ensuite elle permet d'éliminer tous les sous- problèmes qui possèdent une mauvaise évaluation par rapport à la valeur optimale.

- La profondeur d'abord :

Cette stratégie privilégie les sommets les plus éloignés de la racine (de profondeur la plus élevée), en appliquant plus de séparations au problème initial. Cette méthode conduit rapidement à des solutions optimales et économes en mémoire.

3 Conclusion

Dans ce chapitre, nous avons vu plusieurs méthodes de résolution d'un problème d'ordonnancement. Et nous avons dit qu'en fonction de la taille du problème, on distingue deux classes de méthodes pour résoudre un problème donné, les méthodes exactes et approchées.

Ce chapitre est divisé en deux parties, dans la première nous avons abordé quelques notions sur les méthodes approchées qui sont des méthodes utilisées pour résoudre les problèmes avec grand de taille, elle permet de trouver de manière rapide une solution réalisable à un problème donné. Cependant cette solution n'est pas forcément la solution optimale. On distingue deux grandes classes de méthodes approchées les méthodes heuristiques et méta-heuristiques.

Et la deuxième partie interprète les méthodes exactes pour résoudre les problèmes de petite taille. Les méthodes exactes permettent de trouver de manière exacte la solution d'un problème d'optimisation combinatoire. En basant sur la recherche arborescente et sur l'énumération partielle de l'espace de solutions souvent de manière implicite. Nous avons cité nombreuses classe de ces méthodes, notamment la méthode de programmation dynamique qui sera utilisée dans notre travail.

CHAPITRE III :

Expérimentation

1 Introduction

Comme nous avons déjà décrit dans le chapitre précédent, plusieurs méthodes de résolution d'un problème d'ordonnancement existent.

Les méthodes approchées qui permettent de trouver de manière rapide une solution réalisable à un problème donné, et les méthodes exactes (appelées aussi complètes) produisent une solution optimale pour une instance de problème d'optimisation donné. Elles se reposent généralement sur la recherche arborescente et sur l'énumération partielle de l'espace de solutions.

Notre allons présenter dans ce chapitre l'implémentation d'une méthode exacte qui est la programmation dynamique pour résoudre le problème d'allocation des spectres dans la Radio Cognitive.

Nous avons aussi implémenté une métaheuristique qui est l'algorithme génétique pour justifier nos résultats.

2 Environnement de travail

2.1 Environnement matériel

Pour la réalisation de programme, nous avons déployé :

- Un pc portable pour le développement ayant les caractéristiques suivantes :
 - Intel Core i3.
 - 4 Go de mémoire vive.
 - Windows 10 - 64 bits.

2.2 Environnement logiciel

Le langage de développement choisit est java. On a utilisé l'environnement NetBeans pour simplifier le travail afin de développer notre programme. L'environnement NetBeans adopte l'IDE (Integrated Développement Environment) java gratuitement. L'IDE est un logiciel open source développé par SUN Microsystems utilisé pour exécuter, compiler. Un IDE est dédié à un seul langage de programmation.

Le choix se fait pour les raisons suivantes :

- Les logiciels écrits dans ce langage sont très facilement portables sur plusieurs systèmes d'exploitation.
- NetBeans IDE est un outil gratuit et flexible.
- Nous souhaitons, à travers ce projet, améliorer notre connaissance du langage.

3 Présentation du problème

L'énoncé de ce problème est simple : « Étant donné plusieurs utilisateurs secondaires possédant chacun un nombre de canaux et un prix d'achat, étant donné un nombre de canaux maximum autorisé par la radio cognitive (le nombre des canaux non occupés par l'utilisateur principal).

Le problème est quel utilisateur secondaire faut-il rejoindre dans la radio cognitive d'une manière à avoir le meilleur gain sans dépasser le nombre de canaux maximal autorisé.

Dans notre cas,

Nous avons dans un radio cognitive un nombre de canaux maximal "capacité" et n utilisateurs secondaires SU. Pour chaque utilisateur secondaire i, nous avons un nombre de canal "nbrcanaux" et un prix "valeur".

Pour huit utilisateurs (n = 8) et la radio cognitive d'un nombre de canaux maximal autorisé de 15 (capacité = 15), nous avons par exemple les données suivantes :

Num utilisateur	Nombre de canal	Prix
Su 1	2	5
Su 2	3	8
Su 3	5	14
Su 4	2	6
Su 5	4	13
Su 6	6	17
Su 7	3	10
Su 8	1	4

Tableau 2 : le nombre de canal et prix associé aux différents utilisateurs secondaire.

Ensuite, il nous faut définir les variables qui représentent en quelque sorte les actions ou les décisions qui amèneront à trouver une solution.

On définit l'objet SU_utili associée à un utilisateur i

Par exemple pour « $i=0$ » on a :

SU_utili.num = Su 1
SU_utili.nbrcanaux = 2
SU_utili.valeur = 5

Ensuite, on définit la variable xi associée à l'utilisateur i Comme suit : xi = 1 si l'utilisateur i rejoint le spectre de la radio cognitive, et xi = 0 si non.

Dans notre exemple, une solution réalisable est définît comme suit :

$x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 0, x_6 = 0, x_7 = 0$ et $x_8 = 0$.

Puis il faut définir **les contraintes du problème**. Dans notre problème, il n'y a qu'une : la somme des nombres de canal de tous les utilisateurs secondaires alloués doivent être inférieure ou égale au nombre des canaux maximal libre de la radio cognitive.

Pour n utilisateurs, les contraintes peuvent être modéliser comme suit :

$$\sum_{i=1}^{i=n} x_i \cdot \text{nbrcanaux}(i) \leq \text{capacite}$$

La fonction objectif : maximise le gain total .

$$\max \sum_{i=1}^{i=n} xi.valeur (i)$$

4 Méthodes de résolution

Il existe deux grandes catégories de méthodes de résolution de problèmes d'optimisation combinatoire : les méthodes exactes et les méthodes approchées.

Une méthode approchée a pour but de trouver une solution avec un bon compromis entre la qualité de la solution et le temps de calcul.

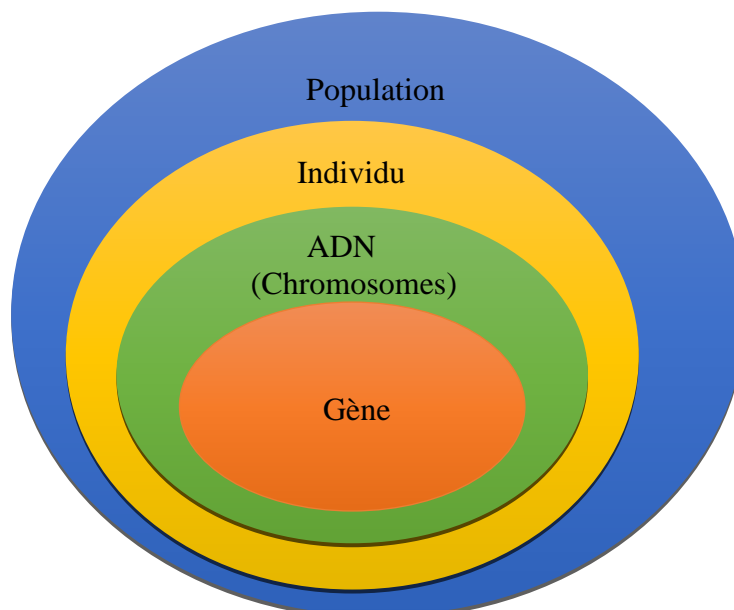
Une méthode exacte a pour but trouver la solution optimale, elle demande un temps de calcul beaucoup plus long (si le problème est difficile à résoudre).

Dans notre projet de fin d'étude On va utiliser l'algorithme une méthode exacte qui est la programmation dynamique. une Méta-heuristique qui est l'algorithme génétique est utilisé pour justifier nos résultats.

4.1 Les algorithmes génétiques

Parmi tous les types d'algorithmes existants, certains ont la particularité de s'inspirer de l'évolution des espèces dans leur cadre naturel. Ce sont les algorithmes génétiques (AGs) sont des algorithmes de recherche stochastiques qui ont été développés pour résoudre les problèmes d'optimisation selon le principe d'évolution des espèces de Charles Darwin.

Ce modèle ci-dessous, lié à la terminologie de la génétique, nous permet d'exploiter les algorithmes génétiques : Nous retrouvons les notions de Population, d'Individu, de Chromosome et de Gène.



La population	→	est l'ensemble des solutions envisageables.
L'individu	→	représente une solution.
Le Chromosome	→	est une composante de la solution.
Le Gène est une	→	caractéristique, une particularité.

Un algorithme génétique va faire évoluer une population dans le but d'en améliorer les individus. Et c'est donc, à chaque génération, un ensemble d'individus qui sera mis en avant et non un individu particulier. Nous obtiendrons donc un ensemble de solutions pour un problème et pas une solution unique.[17]

4.2 Fonctionnement de l'algorithme génétique

Le Fonctionnement d'un algorithme génétique est simple, et leur déroulement peut être découpé en cinq parties :

1. La création de la population initiale ;
2. L'évaluation des individus ;
3. La création de nouveaux individus ;
4. L'insertion des nouveaux individus dans la population ;
5. Répétition du processus.

4.2.1 La création de la population initiale

Pour démarrer un algorithme génétique, il faut lui fournir une population à faire évoluer. Chacun des individus de cette population de base générée aléatoirement Il suffit que tous les individus créés soient de la forme d'une solution potentielle, et il n'est nullement besoin de songer à créer de bons individus. Ils doivent juste rentrer dans le 'moule' du problème posé.

Plus les individus de la population de départ seront différents les uns des autres, plus nous aurons de chances d'y trouver, non pas la solution parfaite, mais de quoi fabriquer les meilleures solutions possibles.

La taille de la population de base est également laissée au programmeur. Une taille de 100 ou 150 individus s'avèrera souvent amplement suffisante, Ensuite, libre à vous de modifier la taille de la population initiale en fonction du problème à résoudre si les solutions trouvées ne vous conviennent pas

4.2.2 Codage et population initiale

Dans un algorithme génétique, un des facteurs les plus importants, si ce n'est le plus important, est la façon dont sont codées les solutions, (c'est-à-dire les structures de données qui coderont les gènes des chromosomes).[8]

Il existe de nombreuses formes de codage, nous citons ici qui on a utilisé dans notre PFE le Codage binaire.

Le principe du codage binaire est de coder la solution selon une chaîne de bits, qui peuvent prendre les valeurs **0** ou **1**. Dans un problème avec une seule variable de décision, par exemple **x**, la valeur de cette variable est transformée en une suite binaire.

Dans notre problème, nous allons coder la solution selon l'utilisateur s'il a rejoint par radio cognitive ou non. Si oui, alors le bit est **1** sinon le bit est **0**.

Exemple :

CHAPITRE III : Expérimentation

Nous choisissons une solution réalisable parmi les solutions possibles de notre problème, par exemple, nous prendrons les utilisateurs colorisés dans les tableaux ci-dessous :

Résultat 1 :

Num utilisateur	Nombre de canal	Prix
Su 1	2	5
Su 2	3	8
Su 3	5	14
Su 9	2	6
Su 10	4	13
Su 11	6	17
Su 12	3	10
Su 13	1	4

Tableau 3 : la première sélection d'utilisateurs secondaires.

Résultat 2 :

Tableau 4 : la 2 eme sélection d'utilisateurs secondaires.

Num utilisateur	Nombre de canal	Prix
Su 1	2	5
Su 2	3	8
Su 3	5	14
Su 4	2	6
Su 5	4	13
Su 6	6	17
Su 7	3	10
Su 8	1	4

Résultat 3 :

Num utilisateur	Nombre de canal	Prix
Su 1	2	5
Su 2	3	8
Su 3	5	14
Su 4	2	6
Su 5	4	13
Su 6	6	17
Su 7	3	10
Su 8	1	4

Tableau 5 : la 3eme sélection d'utilisateurs secondaires

Le codage résultats :

	Su 1	Su 2	Su 3	Su 4	Su 5	Su 6	Su 7	Su 8	
Résultat 1	0	1	0	1	0	1	0	1	→ 35
Résultat 2	1	0	1	1	0	0	1	0	→ 35
Résultat 3	0	1	0	0	0	1	1	1	→ 39

Tableau 6 : codage des trois résultats.

4.2.3 L'évaluation des individus

Une fois que la population initiale a été créée, nous allons en sortir les individus les plus prometteurs, ceux qui vont participer à l'amélioration de notre population. Nous allons donc attribuer une 'note' ou un indice de qualité à chacun de nos individus. La méthode d'évaluation des individus est laissée au programmeur en fonction du problème qu'il a à optimiser ou à résoudre.

4.2.4 La création de nouveaux individus

4.2.4.1 La sélection

Nous allons essayer de prendre des morceaux de solution de certains individus et d'autres morceaux d'autres individus pour créer des nouveaux individus qui, on l'espère, seront des solutions meilleures à notre problème.

Il est tout à fait possible de choisir des individus au hasard et de les mélanger aléatoirement pour créer de nouveaux individus. Ou utilisé des méthodes de sélections : la roulette, la sélection par rang, la sélection par tournoi et l'élitisme.

Dans notre PFE on a utilisé la méthode de sélection l'élitisme Cette méthode de sélection permet de mettre en avant les meilleurs individus de la population. Ce sont donc les individus les plus prometteurs qui vont participer à l'amélioration de notre population. Cette méthode a l'avantage de permettre une convergence (plus) rapide des solutions, mais au détriment de la diversité des individus. On prend en effet le risque d'écartier des individus de piètre qualité, mais qui aurait pu apporter de quoi créer de très bonnes solutions dans les générations suivantes.[17]

	Su 1	Su 2	Su 3	Su 4	Su 5	Su 6	Su 7	Su 8	
Résultat 1	0	1	0	1	0	1	0	1	→ 35
Résultat 3	0	1	0	0	0	1	1	1	→ 39

Tableau 7 : Sélection les deux meilleurs individus dans la population.

4.2.4.2 Les croisements :

Le croisement, ou enjambement, crossing-over, est le résultat obtenu lorsque deux chromosomes partagent leurs particularités. Les croisements permettent de simuler des reproductions d'individus dans le but d'en créer des nouveaux. Il est tout à fait possible de faire des croisements aléatoires. Toutefois, une solution largement utilisée est d'effectuer des croisements multipoints. Il existe deux méthodes de croisement : simple ou double enjambement. Le simple enjambement Il consiste à choisir un emplacement aléatoirement sur une chaîne, et d'intervertir tous les gènes d'un coté de ce point entre les deux chaînes, comme le montre la figure suivante :

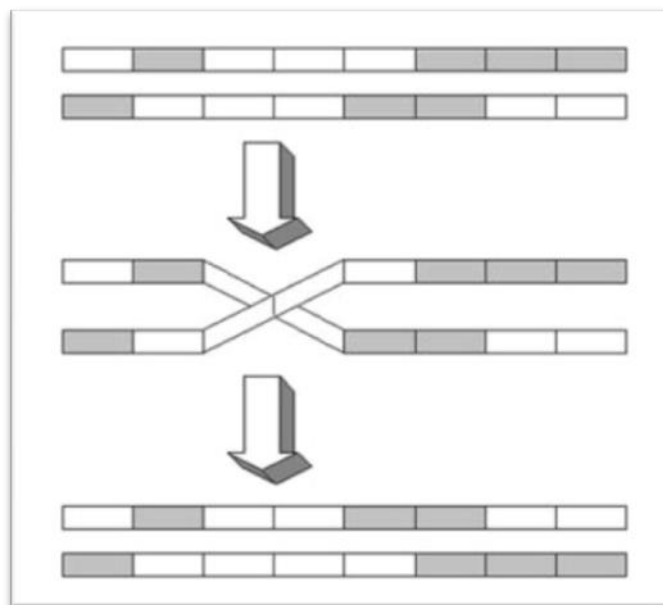


Figure 11 : le croisement des gènes.

Le double enjambement repose sur le même principe, sauf qu'il y a deux points.

Un couple engendre deux enfants dont la première moitié des gènes provient d'un parent et l'autre moitié provient de l'autre parent.

Des individus parents							
Su 1	Su 2	Su 3	Su 4	Su 5	Su 6	Su 7	Su 8
0	1	0	1	0	1	0	1
0	1	0	0	0	1	1	1

Tableau 8 : Les individus parents.

Des individus enfants après le croisement							
Su 1	Su 2	Su 3	Su 4	Su 5	Su 6	Su 7	Su 8
0	1	0	0	0	1	1	1
0	1	0	1	0	1	0	1

Tableau 9 : Les individus enfants.

4.2.4.3 Mutation

Ensuite l'opération de mutation est effectuée sur l'ensemble des enfants obtenues lors du croisement. Pour chacun des enfants, on va supposer qu'un gène est modifié : si c'est un 0, il devient un 1, si c'est un 1 devient un 0. Pour chaque enfant, le gène est choisi aléatoirement.

Il faut de plus s'assurer que la mutation ne crée pas un individu de nombre des canaux supérieur à les canal disponible par la radio cognitive : si c'est le cas, effectuer à nouveau une mutation aléatoire à partir de l'enfant obtenu par croisement jusqu'à ce que l'enfant avec son gène muté soit ce nombre des canaux inférieur ou égal à nombre des canaux disponible par la radio cognitive. Enfin, ajouter les enfants mutés

4.2.5 L'insertion des nouveaux individus dans la population

Une fois que nous avons créé de nouveaux individus que ce soit par croisements ou par mutations, tous les enfants sont évalués. Une politique de remplacement générationnel doit être alors désignée. Une appelée politique élitiste consiste à remplacer dans la génération suivante les moins bons parents par les enfants, alors qu'une autre politique consiste systématiquement à remplacer les parents par leurs enfants dans la population à venir, au risque de perdre certaines bonnes solutions.[8]

4.2.6 Réitération du processus.

Le processus générationnel (sélection, croisement, mutation) est répété jusqu'à ce qu'un critère d'arrêt soit satisfait. Ce critère peut être notamment un nombre maximum de générations, un temps maximal de calcul, une valeur de fitness minimale / maximale, et / ou une convergence vers une solution satisfaisante.

4.3 La programmation dynamique

La programmation dynamique a pour but de déterminer des solutions optimales (maximale ou minimale) à des problèmes possédant la propriété de sous-structures optimales et pour lesquels il y a chevauchement des sous-problèmes. Le chevauchement de sous-problèmes a par exemple été rencontré avec le calcul récursif c'est-à-dire que l'on peut construire la solution optimale du problème à i variables à partir du problème à $i-1$ variables, On recherche dans un premier temps la valeur optimale puis dans un second temps, on en déduit de la valeur optimale la (ou une) solution optimale. L'idée est la suivante :

Étant donné une variable i et le `nbr_canaux` autorisé (c), les solutions optimales de $KP(i,c)$ sont soit:

- Les solutions optimales du problème à $i-1$ variables avec le même `nbr_canaux` c (c.-à-d. $KP(i-1,c)$), auxquelles on ajoute $x_i=0$;
- Les solutions optimales du problème à $i-1$ variables avec le `nbr_canaux` $c - \text{nbr_canaux } i$ (c.-à-d. $KP(i-1,c - \text{nbr_canaux } i)$), auxquelles on ajoute $x_i=1$.
- le problème de zéro utilisateur ($KP(0,*)$) a une solution optimale de valeur nulle.

```

Pour n_canaux de 0 à nbr_canaux_max faire
  T[0,n_canaux] := 0
fin pour

pour i de 1 à n faire
  pour n_canaux de 0 à nbr_canaux_max faire
    si (n_canaux >= nbr_canaux [i]) alors
      T[i,n_canaux] := max(T[i-1,n_canaux], T[i-1, n_canaux -n_canaux [i]] +valeur[i])
    sinon
      T[i,n_canaux] := T[i-1,n_canaux]
    fin si
  fin pour
fin pour
    
```

Num utilisateur	Nombre de canal	Prix
Su 1	2	5
Su 2	3	8
Su 3	5	14
Su 4	2	6
Su 5	4	13
Su 6	6	17
Su 7	3	10
Su 8	1	4

Tableau 10 : le nombre de canal et prix associé aux différents utilisateurs secondaire.

On construit alors un tableau $T[i, n_canaux]$ contenant la valeur des solutions optimales de tout problème $KP(i, n_canaux)$ de la manière suivante :

Nombre de canaux autorisé		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Utilisateur secondaire		Le nombre des canaux libre dans la Radio Cognitive															
Valeur	Nbr_canaux																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	2	0	0	5	5	5	5	5	5	5	5	5	5	5	5	5	5
8	3	0	0	5	8	8	13	13	13	13	13	13	13	13	13	13	13
14	5	0	0	5	8	8	14	14	19	22	22	27	27	27	27	27	27
6	2	0	0	6	8	11	14	14	20	22	25	28	28	33	33	33	33
13	4	0	0	6	8	13	14	19	21	24	27	28	33	35	38	41	41
17	6	0	0	6	8	13	14	19	21	24	27	30	33	36	38	41	44
10	3	0	0	6	8	13	16	19	23	24	29	31	34	37	40	43	46
4	1	0	4	6	10	14	17	20	23	27	29	33	35	38	41	44	47

Tableau 11 : Une matrice des solutions optimales de tout problème $KP(i, nbr_canaux)$

Une fois le tableau construit, pour obtenir la solution optimale il suffit juste de récupérer la valeur de la dernière case de la matrice ($T[n, nbr_canaux]$), et pour récupérer les utilisateurs secondaires de la solution il suffit de démarrer de la case de $T[n, nbr_canaux]$ et de déduire l'état des utilisateurs en remontant jusqu'à une case $T[0, *]$.

5 Experimentation

5.1 L'algorithme génétique

Pour l'algorithme génétique, on doit choisir la meilleure population sur laquelle nous travaillons, généralement le choix de la population initial doit d'être aléatoire avec une mise en jeu d'un seul paramètre qu'est la taille de la population.

5.1.1 Ajustements des paramètres

Nous avons varié la taille de la population avec un pas de 10, commençants par 10 (10 solution réalisable), la meilleure solution trouvée est avec une population de taille 50(voir tableau 11).

Population	10	20	40	50	60	90	100	200	400	700	900
Solution optimale	915	1009	1010	1191	1099	1008	820	913	920	1004	1098

Tableau 12 : le choix de meilleure population

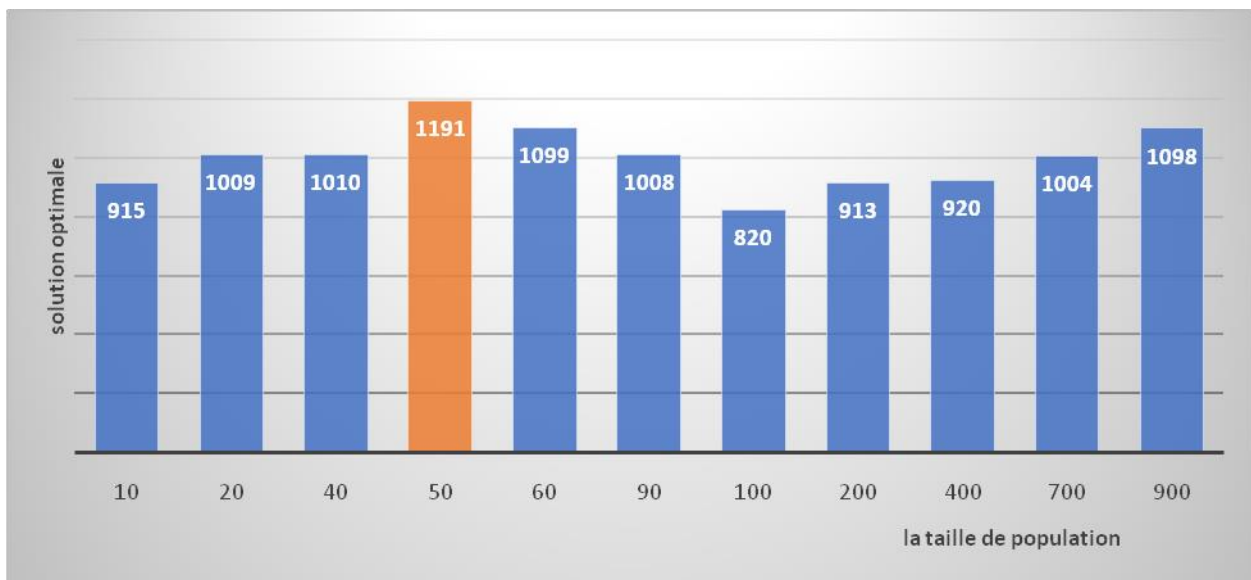


Figure 12 : Histogramme solution optimal par la taille de population.

La fonction **meilleurIndividu** retourner deux meilleurs solution qui appellent les pères.

```

public int[][] meilleurIndividu(int m[][])
{
    int n[][]=new int[2][m[0].length+1];
    int pmax=0;int pmaxl=0;
    int indexpmax=0;
    int indexpmaxl=0;
    for(int i=0;i<m.length;i++)
    {
        //System.out.println("i="+i);
        if((pmax <m[i][m[0].length-1])) {
            indexpmax=i; pmax=m[i][m[0].length-1];
        }
    }

    for(int i=0;i<m.length;i++)
    {
        //System.out.println("i="+i);
        if((pmaxl <m[i][m[0].length-1]) && (m[i][m[0].length-1]!= pmax)) {
            indexpmaxl=i;
            pmaxl=m[i][m[0].length-1];
        }
    }
}

```

Figure 13 : Fonction meilleur Individu.

La fonction **croisement** permet de croiser les gènes des pères à travers d'un point qui était choisie d'une manière aléatoire.

```

// utiliser la methode de crossover -----one point crossover
// cad choisir un point
// apres ce point en fait permutation dans les bits entre les deux pere
public int[][] crossover(int pere[][],ArrayList<S_utili> listeSU,int capaciti)
{
    // la matrice des enfants (2 enfants)
    int Enfants_C[][]=new int[2][pere[0].length];
    Enfants_C[0][Enfants_C[0].length-1]=pere[0][pere[0].length-1];
    Enfants_C[1][Enfants_C[0].length-1]=pere[1][pere[1].length-1];

    // choix d un nombre alatoire pour faire la permutation entre les bite de deux peres
    Random rand = new Random(); //instance of random class
    int upperbound =pere[0].length-2;
    int r = rand.nextInt(upperbound);

    // afin de ginirer enfants
    int pmax=0, cmaxl=0, cmax2=0;
    for(int i=0;i<Enfants_C.length;i++)
    {
        pmax=0;
        cmaxl=0;
        cmax2=0;
        for(int j=0;j<Enfants_C[0].length-2;j++)
        {
            if(j<r)
            {
                Enfants_C[i][j]=pere[i][j];
            }
            else

```

Figure 14 : Fonction de croisement.

Après le croisement des pères on obtient les enfants qui sont des entré d'une autre fonction qu'est la mutation cette dernier permet de modifier la valeur d'un indice, qui était choisie d'une manière aléatoire (1 à 0). Après la modification, le meilleur enfant va remplacer le mauvais père et on le met dans la population et recommencer le traitement jusqu'au ce que la valeur maximale ne change pas après certaine nombre d'itération.

Contrainte :

La condition la plus important dans la fonction de croisement et mutation est que toujours le nombre de canaux dans une solution doit être inférieur ou égale au nombre des canaux libre dans la radio cognitive.

```
public int mutation(int Solution[][],int Enfants_C[][],ArrayList<S_utili> listeSU,int capaciti)
{
    int max=0;

    Random rand = new Random(); //instance of random class
    int upperbound =15;

    // afin de ginirer enfants
    int pmax=0;
    int cmax1=0;
    int cmax2=0;
    //remplire l'enfants l  etrepiter jusqu a cmax1<capaciti

    do{

        int r = rand.nextInt(upperbound);

        //System.out.println(" mutation enfantsl dans le bits  =" +r);
        cmax1=0 ;

        if(Enfants_C[0][r]==1)
        {
            Enfants_C[0][r]=0;

            Enfants_C[0][Enfants_C[0].length-2]= Enfants_C[0][Enfants_C[0].length-2]- (int)listeSU.get(r).getValeur();
        }
        else {
            Enfants_C[0][r]=1;
            Enfants_C[0][Enfants_C[0].length-2]= Enfants_C[0][Enfants_C[0].length-2]+ (int)listeSU.get(r).getValeur();
        }
    }
}
```

Figure 15 : Fonction mutation.

Cette figure (19) donne une simple explication sur le déroulement de l'algorithme génétique.

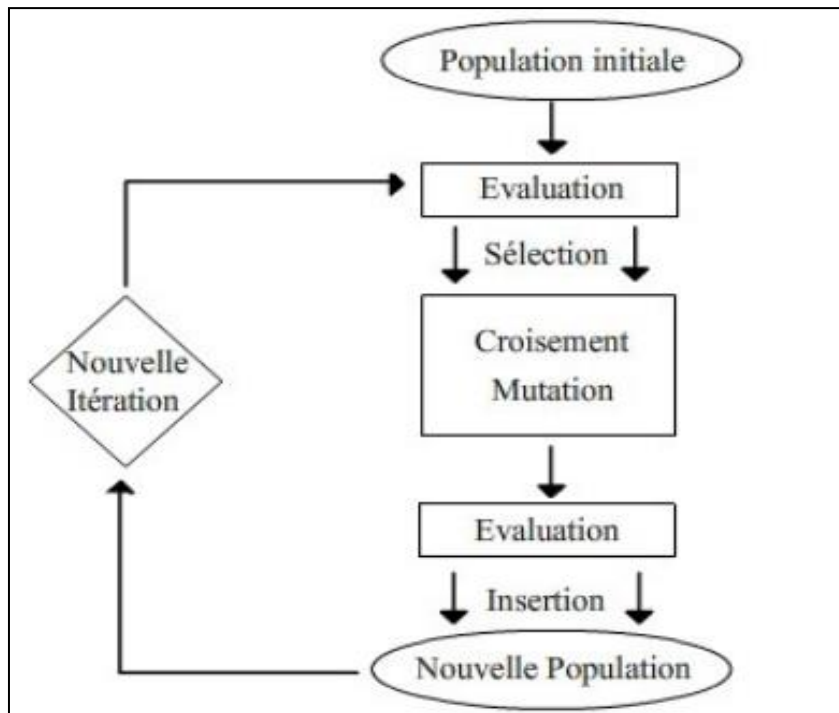


Figure 16 : Fonctionnement de la méthode génétique.

Après le choix de la meilleure population on doit varier les deux variables le nombre des utilisateurs secondaires (nbr_su) et le nombre des canaux libre dans la radio cognitive (capacité). Les résultats sont comme suit :

Scénario	1	2	3	4	5	6	7	8	9	10
Nombre d'utilisateur	100	200	400	600	800	1000	1200	1500	1800	2000
Nombre de canaux autorisé	200	400	600	800	1000	1200	1600	2000	3000	4000
Le gain	1280	2108	2849	3500	4150	4780	5470	6229	8219	10387
Le temps(ms)	6	37	42	66	65	70	88	161	194	130

Tableau 13 : Les résultats des 10 scénarios de la méthode génétique.

5.2 La programmation dynamique :

Pour la programmation dynamique on a deux attributs principale la liste d'utilisateur secondaire qui a une valeur et un nombre des canaux, et le nombre de canaux libre dans la radio cognitive. Le programme va construire une matrice T de M*N tel que les ligne sont des utilisateurs secondaires et les colonnes sont des nombres des canaux qui inclémente de 0 a le nombre des canaux libre par la radio cognitive et T[i][j] c'est le gain.

```

public class Programation_dynamic {

    public int pro2(ArrayList<S_utili> l,int n,int capaciti)
    {int max=0;
      int c=capaciti;
      int[][] T=new int[n+2][c+2];
      for(int i=0;i<c+2;i++)
      {T[0][i]=0;
        //System.out.print( "   "+T[0][i]);
      }
      //System.out.println("");
      for(int i=1;i<n+2;i++)
      {for(int k=0;k<c+1;k++)
        {
          if((k>l.get(i-1).getnbrcanaux())|| (k==l.get(i-1).getnbrcanaux()))
            {T[i][k]=Math.max( T[i-1][k],T[i-1][k-l.get(i-1).getnbrcanaux()]+(int)l.get(i-1).getValeur() );
          }
          else
          {
            T[i][k]=T[i-1][k];
          }
        }
      }
    }
}
    
```

Figure 17 : Fonction de la programmation dynamique.

On a varié le nombre d'utilisateur et le nombre de canaux libre dans la radio cognitive 10 fois et on a obtenu les résultats suivants :

Scénario	1	2	3	4	5	6	7	8	9	10
Nombre d'utilisateur	100	200	400	600	800	1000	1200	1500	1800	2000
Nombre des canaux autorisé	200	400	600	800	1000	1200	1600	2000	3000	4000
Le gain	1280	2200	2940	3590	4240	4800	7270	8480	11280	13720
Le temps(ms)	23	50	78	100	117	126	174	182	223	235

Tableau 14 : Les résultats des 10 scénarios de la programmation dynamique.

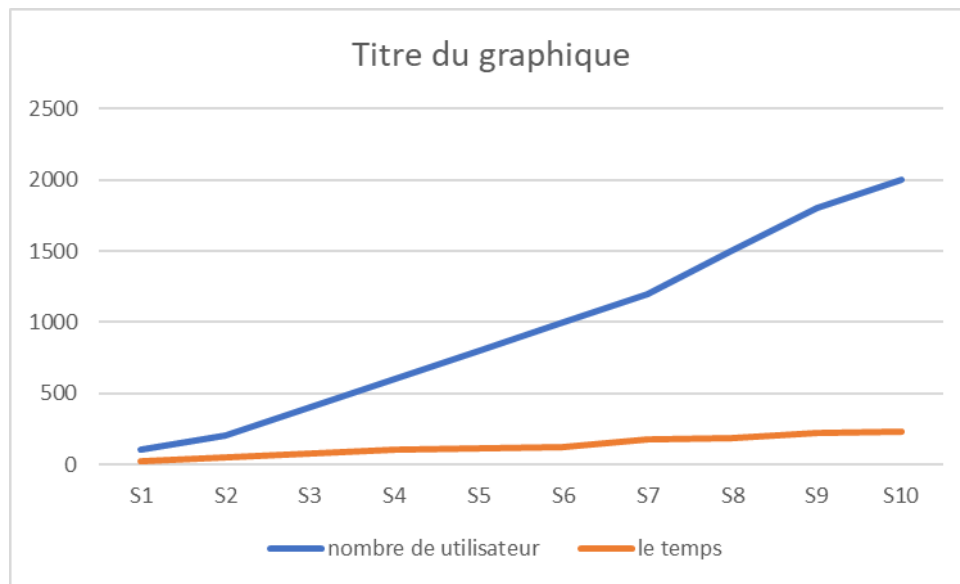


Figure 18 : Graphe de temps d'executions par rapport à le nombre d'utilisateur.

A travers le graphique présenté ci-dessus, on remarque que plus le nombre d'utilisateurs diminue, plus on obtient les résultats en peu de temps, et plus on augmente le nombre d'utilisateurs, plus il faut de temps pour obtenir les résultats. Nous concluons que les deux courbes ont une relation directe proportionnelle. Et nous concluons qu'à un certain moment que la méthode programmation dynamique ou exacte en générale devient inutile et nous devons utiliser les méthodes approché.

6 Comparaison

Les tests qu'on a vue dans les tableaux précédant sont exécutés par les deux programmes avec les même entrés, on a stocké les résultats dans les tableaux suivants afin de comparer les deux méthodes selon les deux critère le temps d'exécution et le résultat obtenue.

6.1 Scénario des tests

Pour l'exécution de programme on a comme entré deux attributs le nombre d'utilisateurs secondaire et le nombre des canaux libre de la radio cognitive, par exemple on prend l'exemple 8 de tableau précédant (tableau 14) on a le nombre d'utilisateur 1500 avec 2000 canaux libre. On a fixé ces donné avec les deux méthodes et lancer l'exécution des programmes, à la fin on a récupéré et affiché les résultats obtenus en terme de gain et e temps d'exécution de ces deux méthodes.

6.2 Comparaison des résultats obtenus en termes de gain:

Scénario	1	2	3	4	5	6	7	8	9	10
La valeur de AG	1280	2108	2849	3500	4150	4780	5470	6229	8219	10387
La valeur de PD	1280	2200	2940	3590	4240	4800	7270	8480	11280	13720

Tableau 15:les résultats de la programmation dynamique et l'algorithme génétique.

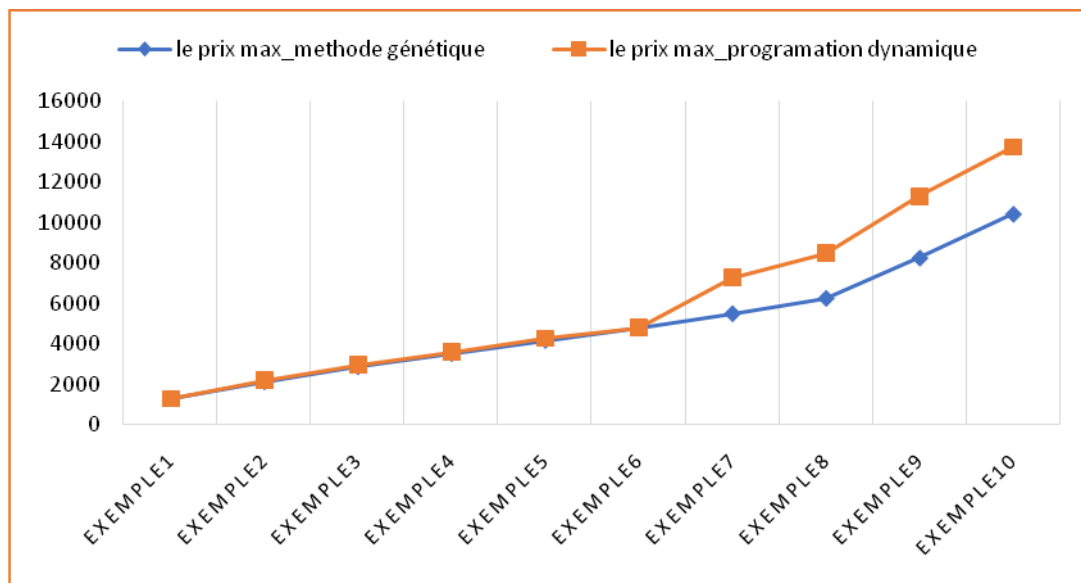


Figure 19 : Histogramme de résultats de la programmation dynamique et l'algorithme génétique

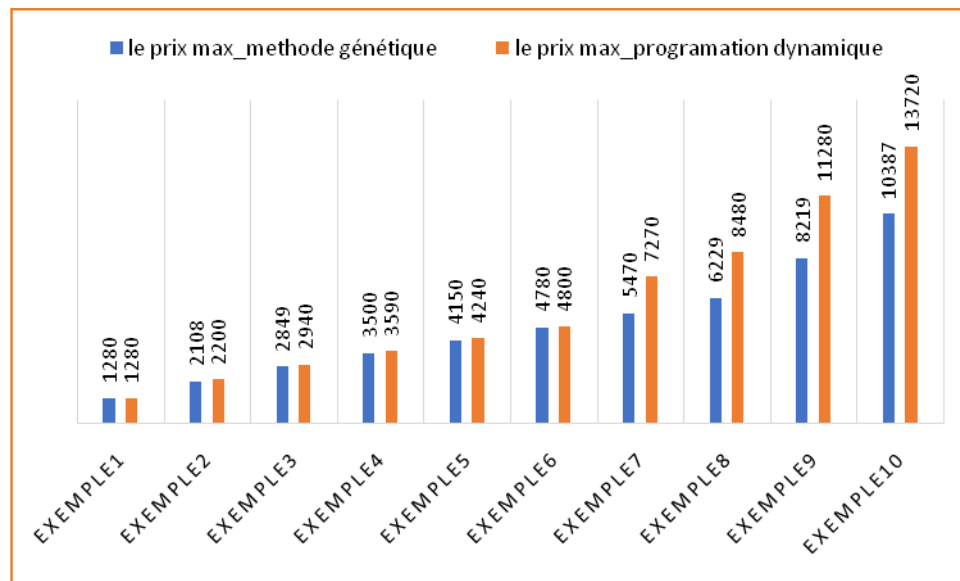


Figure 20 : Le graphe de résultats de la programmation dynamique et l’algorithme génétique.

L’étude comparative réalisée entre ces deux algorithmes proposés en termes de gain montre que la programmation dynamique toujours donne la solution optimale par rapport à l’algorithme génétique qui donne une solution approchée à l’optimalité.

Donc la programmation dynamique c’est la meilleure en termes d’optimalité par rapport à l’algorithme génétique.

6.3 Comparaison des résultats obtenus en termes de temps d’exécution :

Scénario	1	2	3	4	5	6	7	8	9	10
Le temps AG(ms)	6	37	42	66	65	70	88	161	194	130
Le temps PD(ms)	23	50	78	100	117	126	174	182	223	235

Tableau 16 :Le temps d’exécution de la programmation dynamique et l’algorithme génétique .

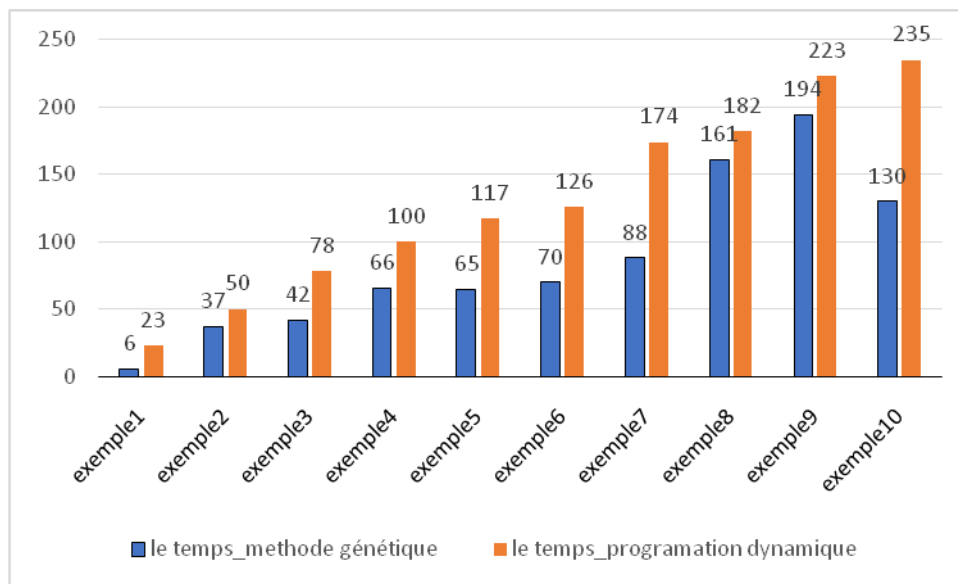


Figure 21 : Histogramme de temps d'exécution de programmation dynamique et l'algorithme génétique

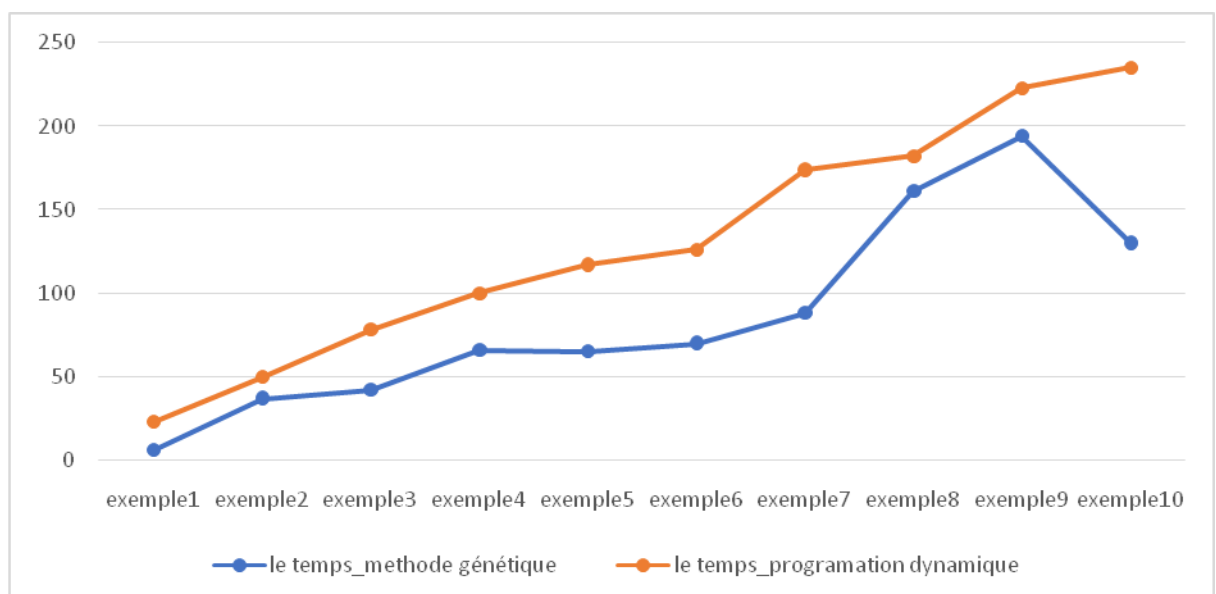


Figure 22 : le graphe de temps d'exécution de la programmation dynamique et l'algorithme génétique

L'étude comparative réalisée entre les deux algorithmes proposés montre que la programmation dynamique prend un temps d'exécution plus long que l'algorithme génétique malgré que les deux fonctions sont exécuté avec les mêmes entrés. Surtout l'hors du passage à l'échelle.

Donc on peut conclure que l'algorithme génétique est meilleure que la programmation dynamique en terme de temps d'exécution.

7 Présentation de l'application :



Figure 23 : Page d'accueil d'application.

L'IHM suivante présente la page d'accueil de notre application, Le bouton « **Start** » lance l'application. En cliquant sur « **Start** », nous affichons l'IHM de notre comparaison s'affiche, nous retrouvons trois champs de saisie de type JTextField chaque champ est affecté à l'un des paramètres des méthodes :

- Capacité : c'est le nombre de canaux autoriser par la radio cognitive
- Random : pour améliorer le résultat de la méthode génétique.
- Nbr_itération : c'est le critère d'arrête de la méthode génétique.

Ainsi qu'une liste de type JComboBox qui contient une liste dont nous allons définir les scénarios des tests.

La validation se fait en cliquant sur le bouton « Programmation Dynamique » ou « Algorithme Génétique » afin d'afficher les résultats



Figure 24 : Interface des résultats.

8 Conclusion

Durant ce chapitre nous avons implémenté une méthode exacte qui est la programmation dynamique pour la gestion des spectres dans les radios cognitives, on a comparé les résultats avec une métaheuristique qui est l'algorithme génétique. Nous remarquons que la programmation dynamique donne la meilleure solution avec un inconvénient de temps d'exécution qui long par rapport à la méthode approchée.

CONCLUSION GENERALE

CONCLUSION GENERALE :

Dans ce projet de fin d'études, on a présenté la radio cognitive dans premier chapitre, la radio cognitive est un nouveau modèle de communication sans fil, dans laquelle un émetteur-récepteur peut détecter intelligemment quels canaux de communication sont utilisés et lesquels ne le sont pas.

L'émetteur-récepteur se déplace alors instantanément dans les canaux libres, tout en évitant les occupés. Ces capacités permettent d'optimiser l'utilisation du spectre de radio fréquences (RF) disponible. la radio cognitive minimise les interférences avec les autres utilisateurs. Et, en évitant les canaux occupés, il augmente l'efficacité du spectre et améliore la qualité de service pour les utilisateurs.

Dans le deuxième chapitre on a parler sur les algorithmes de résolution des problèmes d'optimisation combinatoire qui sont classés en deux grandes catégories :Les méthodes exactes et les méthodes approchées, cette dernière permet de trouver des solutions avec un temps de calcul raisonnable mais on obtient une solution approchée d'un problème donné par contre à la méthode exacte qui donne une solution optimale mais dans un temps plus longue que la méthode approchée. Nous avons ensuite cité quelques algorithmes de méthode approchée et exacte par exemple Branch and Bound, L'algorithme A*, La génération de colonnes et La programmation dynamique sont des algorithmes de la classe exacte, et Méthode méta-heuristique, La méthode de descente et L'algorithme glouton sont des méthodes approchées.

Et pour finir notre projet de fin d'études, au troisième chapitre nous avons créé deux programmes, l'un par la programmation dynamique et l'autre on a utilisé l'algorithmes génétique afin de résoudre le problème d'allocation des ressources dans les réseaux radio cognitifs, et à la fin nous avons comparé les résultats obtenus en termes de l'optimalité et en termes de temps d'exécution. A travers une série d'expérimentations sur des problèmes différents (le nombre d'utilisateur secondaire et le nombre des canaux libre) nous avons montré que l'algorithme de la programmation dynamique est efficace et donne des solutions optimales mais dans un temps long par contre l'algorithme génétique donne une solution réalisable approchée à l'optimalité dans un temps raisonnable.

Comme perspective de ce travail, il est possible d'utiliser l'algorithme génétique pour générer la solution initiale afin d'assurer la rapidité de la méthode de la programmation dynamique pour augmenter la performance de cette méthode.

CONCLUSION GENERALE

Résumé

La communication sans fils a rencontré plusieurs problèmes d'utilisation de la ressource spectrale, une nouvelle approche qui est la radio cognitive propose des techniques intelligent pour résoudre le problème d'allocation des ressources afin d'améliorer l'utilisation de spectre et éviter les interférences, Pour cela nous proposons dans ce PFE la programmation dynamique comme méthode exacte pour l'allocation des ressources dans les radios cognitives. Nous avons justifié nos résultats par une comparaison avec une métaheuristique qui est l'algorithme génétique, les comparaisons sont faite selon deux critère important qui sont le temps d'exécution et le gain.

Mots clé: radio cognitive, allocation de ressource, méthode exacte ,méthode approchée ,programmation dynamique ,algorithmes génétique.

Abstract

Wireless communication has encountered several problems of spectrum resource utilization, a new approach which is cognitive radio offers intelligent techniques to solve the resource allocation problem in order to improve spectrum utilization and avoid interference. , For this we propose in this work dynamic programming as an exact method for the allocation of resources in cognitive radio.

We justified our results by a comparison with a metaheuristic which is the genetic algorithm, the comparisons are made according to two important criteria which are the execution time and the gain.

Keywords: cognitive radio, resource allocation, exact method, approximate method, dynamic programming, genetic algorithms

ملخص

واجه الاتصال اللاسلكي العديد من مشاكل استخدام موارد الطيف ، نهج جديد وهو الراديو المعرفي يقدم تقنيات ذكية لحل مشكلة تخصيص الموارد من أجل تحسين استخدام الطيف وتجنب التداخل. طريقة لتخصيص الموارد في الراديو الاستعرافي.

لقد بررنا نتائجنا من خلال المقارنة مع metaheuristic وهي الخوارزمية الجينية ، ويتم إجراء المقارنات وفقاً لمعيارين مهمين وهما وقت التنفيذ والمكاسب.

الكلمات المفتاحية: الراديو المعرفي، تخصيص الموارد، الطريقة الدقيقة، الطريقة التقريبية، البرمجة الديناميكية ، الخوارزميات الجينية

Bibliographie

- [1] Badr Benmammar. Présentation de la radio cognitive. 3rd cycle. 2012. ffccl-00680189v2
- [2] Blog-qu-est-ce-qu-une-radio-logicielle. [Consulter : 05-07-2022.] <https://www.syrlinks.com/fr/blog-qu-est-ce-qu-une-radio-logicielle>.
- [3] Internet des Objets. [Consulter : 06- 07-2022.] <https://blog.rtone.fr/sdr>.
- [4] Federal Communications Commission, First Report and Order in ET Docket. 2001.
- [5] Badr Benmammar et Asma Amraoui. Réseaux de radio cognitive Allocation des ressources radio et accès dynamique au spectre. s.l. : LTT Laboratoire de Télécommunications Tlemcen, UABT, Algérie.
- [6] Qu'est-ce que la radio cognitive? - définition de techopedia. [Consulter 12-07- 2022] <https://fr.theastrologypage.com/cognitive-radio>.
- [7] Taleb Mohammed Housseyn. implementation-du-clustering-dans-un-reseau-de-radio-cognitive-en-utilisant-la-plate-forme-jade. projet de fin d'etudes . DEPOT institutionnel de l'Université Abou Bekr Belkaid Tlemcen.2016
- [8] Benghabrit Nawel. Allocation des ressources dans les réseaux radio.Univ. Oran 1 Ahmed Ben Bella. oran : s.n., 19 Mars 2019.
- [9] Nada kherici. méthodes de résolution en optimisation. Cours Optimisation Combinatoire UBMA, 2020/2021
- [10] Bouyerden Amina et Hecham Fatima Zohra:Application des méthodes exactes dans les réseaux de radio cognitive.Mémoire de fin d'études
- [11] Algorithmes : Heuristiques et méta-heuristiques. Eurodecision Algorithmes Pour Business.<https://www.eurodecision.com/algorithmes/recherche-operationnelle-optimisation/heuristiques-metaheuristiques>.
- [12] Algo glouton. MLN (monlyceenumerique). http://monlyceenumerique.fr/nsi_premiere/algo_a/a6_algo_gloutons.php.
- [13] Raksmei Phan. Méthodes exactes et approchées par partition en cliques de graphes. Français. : Université Blaise Pascal - Clermont-Ferrand II. 2013.
- [14] AYAD Noureddine. memoire de fin d'etudes :Utilisation des méta-heuristiques pour l'équilibrage des charges dans le Cloud Computing. DEPOT institutionnel de l'Université Abou Bekr Belkaid Tlemcen .2021.
- [15] Djamel Rebaïne: "La méthode de branch and bound". Note de cours
- [16] Sidi Mohamed Douiri, Souad Elbernoussi, Halima Lakhbab.Méthodes de Résolution Exactes Heuristiques et Métaheuristiques. CoursUniversité Mohammed V, Faculté des Sciences de Rabat.
- [17] khayyam. Les algorithmes génétiques. developpez.com. 2005 <https://khayyam.developpez.com/articles/algo/genetic/#Lno-I>.