

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

ALLOCATION DES RESSOURCES DANS LES RÉSEAUX CELLULAIRES DU
FUTUR : ALGORITHMES ET COMPLEXITÉS

THÈSE
PRÉSENTÉE
COMME EXIGENCE PARTIELLE
DU DOCTORAT EN INFORMATIQUE

PAR
ZOUBEIR MLIKA

MAI 2019

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de cette thèse se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.10-2015). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

REMERCIEMENTS

Je tiens à remercier ma chère mère Hedia qui m'a continuellement soutenu et encouragé. Je remercie mon cher père Mohamed qui m'a encouragé et supporté. Je vous remercie beaucoup mes chers parents. Je remercie aussi mon frère Hamdi, ma sœur Fatma, mon cousin Mahdi, ma tante Jamila et mes oncles Rached et Ibrahim. Un grand merci à toute ma famille.

À toute ma famille, même si je ne peux pas prédire l'avenir, je sais que je peux compter sur vous.

Je remercie mon directeur de recherche prof. Wessam Ajib pour son aide, ses encouragements, ses conseils et ses motivations.

Je remercie prof. Elmahdi Driouch qui m'a co-supervisé, a travaillé avec moi et m'a offert son savoir.

Je remercie mon ami Zakaria Elalaoui pour son aide et ses conseils.

Je remercie Amina Hentati pour ses encouragements.

Je remercie Mathew Goonewardena qui a travaillé avec moi pendant ma maîtrise à l'UQAM.

Je tiens à remercier aussi tous mes amis à l'UQAM : Rami, Yosr, Amina, Wael, Cyrine, Syrine, Mohamed, Tri, Tilahun, Françoise, et la liste est longue.

Je remercie tous les professeurs qui m'ont enseigné. Je remercie spécialement les professeurs de l'UQAM : prof. Halima Elbiaze, prof. Guy Bégin, prof. Timothy Walsh, pour nommer quelques-uns.

Un remerciement spécial pour le membre de jury pour l'effort qu'ils ont fait afin d'évaluer la thèse. Je remercie prof. Alexandre Blondin-Massé, prof. Guy Bégin et prof. Abdoulaye Baniré Diallo pour leurs conseils durant ma proposition de thèse.

Un grand merci à tous et à toutes !

TABLE DES MATIÈRES

REMERCIEMENTS	ii
LISTE DES TABLEAUX	ix
LISTE DES FIGURES	xi
LISTE DES ALGORITHMES	xiv
LISTE DES ABRÉVIATIONS ET SYMBOLES MATHÉMATIQUES	xvi
RÉSUMÉ	xx
CHAPITRE I	
INTRODUCTION	1
1.1 Motivations et problématiques	1
1.1.1 Les ressources dans les réseaux DCNs	7
1.2 Objectifs	14
1.2.1 L'objectif général	14
1.2.2 Les objectifs spécifiques	15
1.2.3 Résumés des objectifs spécifiques	17
1.3 Méthodologie générale	18
1.3.1 Définitions	21
1.4 Contributions de la thèse	22
1.4.1 Contribution 1 : association centralisée et distribuée des utilisateurs sous les contraintes de SINR	23
1.4.2 Contribution 2 : activation des stations de base et association des utilisateurs pour une meilleure efficacité énergétique	24
1.4.3 Contribution 3 : ordonnancement, association des utilisateurs et al- location de fréquences sans et avec la récolte d'énergie dans les scé- narios <i>offline</i> et <i>online</i>	25
1.5 Organisation de la thèse	27

CHAPITRE II	
ASSOCIATION CENTRALISÉE ET DISTRIBUÉE DES UTILISATEURS SOUS LES CONTRAINTES DE SINR	29
2.1 Résumé	29
2.2 Introduction	30
2.2.1 Contributions	32
2.2.2 Organisation	32
2.3 Revue de littérature	32
2.4 Modèle de système	35
2.5 Formulation du problème	37
2.6 NP-difficulté	39
2.7 Algorithmes centralisés	42
2.8 Algorithmes distribués	46
2.8.1 Un jeu non-coopératif	47
2.8.2 Algorithmes proposés	50
2.9 Résultats de simulations	58
2.9.1 Résultats pour les algorithmes centralisés	58
2.9.2 Résultats pour les algorithmes distribués	64
2.10 Conclusions	71
CHAPITRE III	
ACTIVATION DES STATIONS DE BASE ET ASSOCIATION DES UTILISA- TEURS POUR UNE MEILLEURE EFFICACITÉ ÉNERGÉTIQUE	73
3.1 Résumé	73
3.2 Introduction	74
3.2.1 Contributions	75
3.2.2 Organisation	76
3.3 Revue de littérature	77
3.4 Modèle de système et formulation du problème	80

3.5	BOA avec débits minima élevés	85
3.5.1	NP-difficulté	85
3.5.2	Solutions proposées	89
3.6	BOA avec débits minima faibles	95
3.6.1	NP-difficulté	97
3.6.2	Solutions proposées	99
3.7	Résultats de simulations	106
3.8	Conclusions	117
CHAPITRE IV		
ORDONNANCEMENT, ASSOCIATION DES UTILISATEURS ET ALLOCA-		
TION DE FRÉQUENCES SANS ET AVEC LA RÉCOLTE D'ÉNERGIE DANS		
LES SCÉNARIOS <i>OFFLINE</i> ET <i>ONLINE</i>		119
4.1	Résumé	120
4.2	Introduction	121
4.2.1	Contributions	122
4.2.2	Organisation	123
4.3	Revue de littérature	123
4.4	Modèle de système	125
4.5	Le problème OAD	129
4.5.1	Formulation du problème	129
4.5.2	NP-difficulté et solutions proposées	131
4.5.3	Résultats de simulation	136
4.6	Le problème OAFED	140
4.6.1	Formulation du problème	141
4.6.2	NP-difficulté et solutions proposées	143
4.6.3	Résultats de simulations	158
4.7	Le problème OAED-ON	162
4.7.1	Le cas d'une seule station de base	165

4.7.2	Le cas de plusieurs stations de base	179
4.7.3	Résultats de simulations	181
4.8	Conclusions	187
CHAPITRE V		
CONCLUSIONS ET TRAVAUX FUTURS		189
5.1	Conclusions	189
5.2	Travaux futurs	194
RÉFÉRENCES		199

LISTE DES TABLEAUX

Tableau		Page
2.1	Complexité et informations échangées des algorithmes distribués pour MSA.	58
2.2	Paramètres de simulations pour les algorithmes distribués pour MSA. .	64
3.1	Liste des notations pour BOA.	84
3.2	Paramètres de simulations pour BOA.	106
4.1	NP-difficulté, algorithmes et complexité asymptotique de OAFED . . .	144
4.2	Paramètres de simulations pour OAFED.	159

LISTE DES FIGURES

Figure	Page
1.1 Prévisions Cisco pour la croissance du trafic de données mobile (Cisco, 2017).	2
1.2 Augmentation du trafic sans fil en Amérique du Nord entre 2007 et 2020 (Andrews <i>et al.</i> , 2012). L'axe des ordonnées mesure la quantité de trafic en téraoctet par seconde (Tb/s).	3
1.3 Augmentation de l'efficacité spectrale en fonction de la densité des réseaux cellulaires (TEKTELIC Communications, 2014).	5
1.4 Illustration du problème d'interférence et d'association des utilisateurs aux stations de base dans les réseaux DCNs. L'idée de cette figure vient d'une figure similaire publiée dans (Lopez-Perez <i>et al.</i> , 2011).	9
1.5 La méthodologie de la thèse.	20
2.1 Nombre moyen d'utilisateurs associés par OPT-MSA et ALG-MSA en fonction de m pour MSA.	60
2.2 Nombre moyen d'utilisateurs associés par ALG-MSA et OPT-MSA en fonction de n pour MSA.	61
2.3 Efficacité spectrale moyenne par OPT-MSA et ALG-MSA en fonction de k pour MSA.	62
2.4 Fonction de répartition complémentaire de l'efficacité spectrale par OPT-MSA et ALG-MSA pour MSA.	63
2.5 L'impact du facteur gagnant τ sur l'algorithme mWSLS pour MSA.	65
2.6 L'impact du facteur perdant ϵ sur l'algorithme mWSLS pour MSA.	66
2.7 L'impact du nombre d'itérations sur l'algorithme mWSLS avec $\epsilon = \epsilon^*$ pour MSA.	67
2.8 Nombre moyen d'utilisateurs associés par les algorithmes distribués en fonction de n pour MSA.	68

2.9	Équilibrage de charge pour les algorithmes mWSLS et LB-mWSLS en fonction de n pour MSA.	69
2.10	Le prix d'anarchie et de stabilité du jeu $\mathcal{G}$ en fonction de n pour MSA. .	70
3.1	Faisabilité de BOAH en fonction de k avec $n = 16$	108
3.2	Faisabilité de BOAL en fonction de k avec $n = 16$	109
3.3	Nombre moyen de stations de base ON par tous les algorithmes en fonction de k pour BOAH.	110
3.4	Nombre moyen de stations de base ON par tous les algorithmes en fonction de k pour BOAL.	111
3.5	Temps d'exécution des algorithmes en fonction de n avec $k = 6$ pour BOAH.	112
3.6	Temps d'exécution des algorithmes en fonction de n avec $k = 20$ pour BOAL.	113
3.7	Fonction de répartition de l'efficacité énergétique par tous les algorithmes pour BOAH.	114
3.8	Fonction de répartition de l'efficacité énergétique par tous les algorithmes pour BOAL.	115
3.9	Temps de séjour moyen pour tous les algorithmes avec $k = 6$, $n = 10$ pour BOAH et BOAL.	116
3.10	Nombre moyen de stations de base ON par tous les algorithmes en fonction de k avec $n = 100$ pour BOAM.	117
4.1	Un exemple du modèle de système où $\ell = 4$ et $f_i = i$	128
4.2	Un exemple d'une solution faisable à (4.5) qui est constituée de deux ordonnanceurs; un pour une macro-cellule et un pour une petite cellule.	131
4.3	Nombre moyen d'utilisateurs associés par tous les algorithmes en fonction de n	137
4.4	Nombre moyen d'utilisateurs associés par OPT-OAD et ALG-OAD en fonction de k	138
4.5	Équité d'utilisateurs par OPT-OAD et ALG-OAD en fonction de k	139

4.6	Efficacité spectrale par tous les algorithmes en fonction de k pour OAD.	140
4.7	Un exemple d'un ordonnanceur Σ_b d'une station de base b .	146
4.8	Une illustration de la démonstration du lemme 4.6.4.	151
4.9	Nombre moyen d'utilisateurs servis par ALG-SFSB en fonction de k pour OAFED-SFSB.	160
4.10	Nombre moyen d'utilisateurs servis par ALG-SFSB en fonction de k pour OAFED-SFSB.	161
4.11	Nombre moyen d'utilisateurs servis par ALG-SFMB et OPT-SFMB en fonction de k pour OAFED-SFMB pour différents λ .	162
4.12	Nombre moyen d'utilisateurs servis par ALG-SFMB et OPT-SFMB en fonction de k pour OAFED-SFMB pour différents m .	163
4.13	Nombre moyen d'utilisateurs servis par ALG-SFMB et OPT-SFMB en fonction de k pour OAFED-SFMB pour différents n .	164
4.14	Nombre moyen d'utilisateurs servis par ALG-MFMB et OPT-MFMB en fonction de k pour OAFED-MFMB pour différents n ($\ell = 2$).	165
4.15	Nombre moyen d'utilisateurs servis par ALG-MFMB pour OAFED-MFMB en fonction de λ pour différents n et ℓ ($k = 100$).	166
4.16	Nombre moyen d'utilisateurs servis par tous les algorithmes pour OED-ON en fonction de T pour $m = 5$.	182
4.17	Nombre moyen d'utilisateurs servis par tous les algorithmes pour OED-ON en fonction λ pour $T = 100$.	183
4.18	Le regret en fonction du temps pour OED-ON ($n = 5$ et $\lambda = 2$).	184
4.19	Le regret en fonction du temps pour OED-ON ($\delta = 20$ et $\lambda = 2$).	185
4.20	Nombre moyen d'utilisateurs servis par tous les algorithmes pour OAED-ON en fonction de n ($m = 10$ et $T = 50$).	186

LISTE DES ALGORITHMES

1	L'algorithme totalement distribué mWSLS pour le joueur (b, s)	52
2	L'algorithme distribué de la meilleure réponse BRD	55
3	L'algorithme distriubé max-SINR	56
4	L'algorithme totalement distribué RWM pour le joueur (b, s)	56
5	L'algorithme ALG-BOAH	94
6	L'algorithme ALG-BOAL	101
7	L'algorithme ALG-OAD	134
8	L'algorithme ALG-SFSB	147
9	La procédure de re-ordonnancement	149
10	L'algorithme ALG-SFMB	154
11	L'algorithme ALG-MFMB	157
12	L'algorithme OFF-OED	167
13	L'algorithme ON-OED appliqué pendant trame t	169
14	L'algorithme ONL-OED	175
15	L'algorithme OFF-OAED	179

LISTE DES ABRÉVIATIONS ET SYMBOLES MATHÉMATIQUES

$\arg \max\{\cdot\}$ Indice du maximum

$\setminus$ Différence

$\binom{\cdot}{\cdot}$ Coefficient binomial

$\cup$ Union

$\exists$ Il existe

$\forall$ Pour tout

$\in$ Appartient à

A Lettre en gras majuscule pour noter une matrice

a Lettre en gras minuscule pour noter un vecteur

$\mathcal{O}$ La notation grand-O

$\max\{\cdot\}$ Maximum de

$|\cdot|$ Cardinal d'un ensemble

$\min\{\cdot\}$ Minimum de

$\sqrt{\cdot}$ Racine carrée

$\subset$ Inclusion

Σ Symbole somme

$\times$ Produit cartésien

$\triangleq$ Égalité par définition

3GPP *3rd generation partnership project*

ALG-X *Algorithme proposé pour problème X*

OPT-X *Algorithme optimal pour problème X*

AWGN *Additive white gaussian noise*

BOA *Base station operation and association*

BOAH *Base station operation and association with high rate requirements*

BOAL *Base station operation and association with low rate requirements*

BOAM *Base station operation and association with mixed rate requirements*

BRD *Best response dynamics*

c.-à-d. *C'est-à-dire*

CAPEX *Capital expenditure*

CRTC *Canadian radio-television and telecommunications commission*

DCNs *Dense cellular networks*

eMBB *evolved mobile broadband*

FCC *Federal communications commission*

GHz *Gigahertz*

GSM *Global system for mobile communications*

Hz *Hertz*

ISED *Innovation, science and economic development canada*

LTE *Long term evolution*

LTE-A *Long term evolution advanced*

MAB *Multi-armed bandit*

MHz *Megahertz*

MMAB *Mortal multi-armed bandit*

MSA *Multi-slot association*

nG *n-th generation*

NP *La classe de complexité non déterministe polynomiale*

NTIA *National telecommunications and information administration*

OAD *Ordonnancement et association avec dates limites*

OAED-ON *Ordonnancement et association avec énergie et dates limites online*

OAFED *Ordonnancement, association et fréquence avec énergie et dates limites*

OED-ON *Ordonnancement avec énergie et dates limites online*

OPEX *Operating expense*

P *La classe de complexité polynomiale*

p. ex. *Par exemple*

resp. *Respectivement*

RWM *Randomized weighted majority*

s *Secondes*

SINR *Signal to interference plus noise ratio*

SNR *Signal to noise ratio*

SSA *Single-slot association*

xx

TDMA *Time division multiple access*

WSLS *Win-stay, lose-shift*

RÉSUMÉ

Les réseaux cellulaires ont connu durant les dernières décennies des avancées technologiques considérables. Plusieurs générations de ces réseaux se sont succédé, et chacune offrait une gamme de plus en plus large d'applications de haute performance aux utilisateurs. La première et la deuxième générations, plus connues sous les sigles 1G et 2G, étaient spécialement conçues pour le transport de la voix. Les générations récentes sont dotées de la capacité de transporter des données; permettant ainsi un accès continu à Internet. Les exigences de la prochaine génération portent encore la barre plus haut avec un débit de transmission plus élevé (allant jusqu'à des dizaine de gigabits par seconde), un délai de latence plus réduit (moins d'une milliseconde) ainsi qu'une couverture et une disponibilité quasi-complète. Afin d'atteindre ces exigences, la prochaine génération adoptera plusieurs nouvelles technologies, telles que les antennes multiples à large échelle, la communication millimétrique, et les réseaux cellulaires denses hétérogènes (appelés *Dense Cellular Networks* ou simplement DCNs). Les réseaux DCNs se caractérisent par le déploiement de plusieurs stations de base de différentes caractéristiques (tailles, puissance, prix, etc.) avec des distances réduites entre elles.

Dans cette thèse, nous proposons des algorithmes d'allocation de ressources dans les réseaux DCNs de la prochaine génération. Spécifiquement, nous étudions le problème d'ordonnancement et d'association des utilisateurs aux différentes stations de base ainsi que l'allocation des fréquences. Nous nous intéressons à (i) étudier la complexité computationnelle de ce problème en caractérisant sa NP-difficulté et (ii) proposer des algorithmes d'allocation de ressources avec des performances théoriques dans le pire cas (comme des algorithmes d'approximations ou des algorithmes compétitifs). Dans la première partie de cette thèse, nous étudions le problème d'association des utilisateurs aux stations de base sous les contraintes d'interférence et de qualité de service avec l'objectif de maximiser le nombre d'utilisateurs associés. Ce problème, à nos connaissances, n'a pas été étudié par d'autres chercheurs. Nous montrons la NP-difficulté du problème et nous proposons un algorithme d'approximation permettant de le résoudre efficacement. Ensuite, nous modélisons ce problème en utilisant la théorie des jeux. Ainsi, nous proposons un algorithme d'apprentissage totalement distribué. En second lieu, nous étudions le problème conjoint d'association des utilisateurs avec activation et désactivation des stations de base. L'objectif est de minimiser la consommation énergétique sous les contraintes de qualité de service des utilisateurs. Nous étudions la faisabilité de ce problème et nous montrons qu'il s'agit d'un problème NP-difficile. Nous proposons ensuite différentes solutions algorithmiques pour différents cas de qualité de service exigée.

Dans la troisième partie de cette thèse, nous étudions le problème conjoint d'ordonnement, d'association des utilisateurs et d'allocation de fréquences avec ou sans la récolte d'énergie. L'objectif est de maximiser le nombre d'utilisateurs satisfaits. Ce problème est résolu sous les contraintes de qualité de service et de délai contraignant. Dans le cas sans récolte d'énergie, nous étudions un cas spécial de ce problème et nous proposons un algorithme optimal pour le résoudre. Après, nous démontrons la NP-difficulté du problème dans le cas général et nous proposons un algorithme d'approximation avec un facteur d'approximation constant. Dans le cas avec récolte d'énergie, nous étudions la NP-difficulté du problème dans différents cas. Lorsque le problème appartient à la classe P, nous proposons un algorithme optimal pour le résoudre. Lorsque le problème est NP-difficile, nous proposons des algorithmes d'approximation ainsi que des algorithmes heuristiques. Ensuite, nous résolvons ce problème dans un scénario *online*. Nous proposons des algorithmes *online* compétitifs ainsi qu'un algorithme d'apprentissage incrémental et nous démontrons que son regret est sous-linéaire. Pour chaque problème étudié dans ce projet, nous présentons des résultats de simulations Monte Carlo afin d'illustrer les performances de nos solutions proposées. La dernière partie de cette thèse conclut le travail et présente différentes pistes de recherche pour le futur.

Mots clés : Réseaux cellulaires denses hétérogènes, allocation de ressources, ordonnancement, association des utilisateurs, allocation de fréquences, récolte d'énergie, activation et désactivation des stations de base, interférence, capacité, délai, dates limites, complexité asymptotique, NP-difficulté, algorithmes optimaux, algorithmes gloutons, algorithmes compétitifs, algorithmes d'approximation, algorithmes d'apprentissage.

CHAPITRE I

INTRODUCTION

Ce chapitre introduit le projet de ma thèse de doctorat. Il est divisé en cinq parties : (i) les motivations et les problématiques, (ii) les objectifs, (iii) la méthodologie, (iv) les contributions, et finalement (v) l'organisation de la thèse.

1.1 Motivations et problématiques

Le nombre d'appareils mobiles a explosé dans les deux dernières décennies et il ne cesse d'augmenter. En effet, dans une étude faite par Cisco (Cisco, 2017), il y aura à peu près 12 milliards d'appareils mobiles d'ici 2021—dépassant la population mondiale prévue à cette époque. En outre, la demande des services sans fil à travers les réseaux cellulaires tels que les données, la voix, la télé-santé, les jeux en ligne est toujours en augmentation (Hossain *et al.*, 2013). Cette démocratisation des appareils mobiles, des tablettes tactiles et d'autres appareils portables a fait augmenter le trafic sans fil rapidement (Cisco, 2017). À titre d'exemple, bien que la communication cellulaire de la quatrième génération (4G) ne représente que 26% de la communication mobile en 2016, elle représentait déjà 69% du trafic de données mobiles. En 2021, la communication 4G représentera 53% de la communication mobile mais elle représentera 79% du trafic de données mobiles (Cisco, 2017). Nous pouvons voir sur la figure 1.1 que le trafic sans fil a augmenté à peu près deux fois et demi entre 2016 et 2018, de 7 exaoctets (EO) par mois à 17 EO par mois—soit avec

un taux de croissance annuel composé (TCAC) de 55%. En 2021, le trafic sans fil atteindra 49 EO par mois, donc il augmentera sept fois par rapport à 2016—soit avec un TCAC de 47%. Une autre étude faite par (Andrews *et al.*, 2012) montre l'augmentation expo-

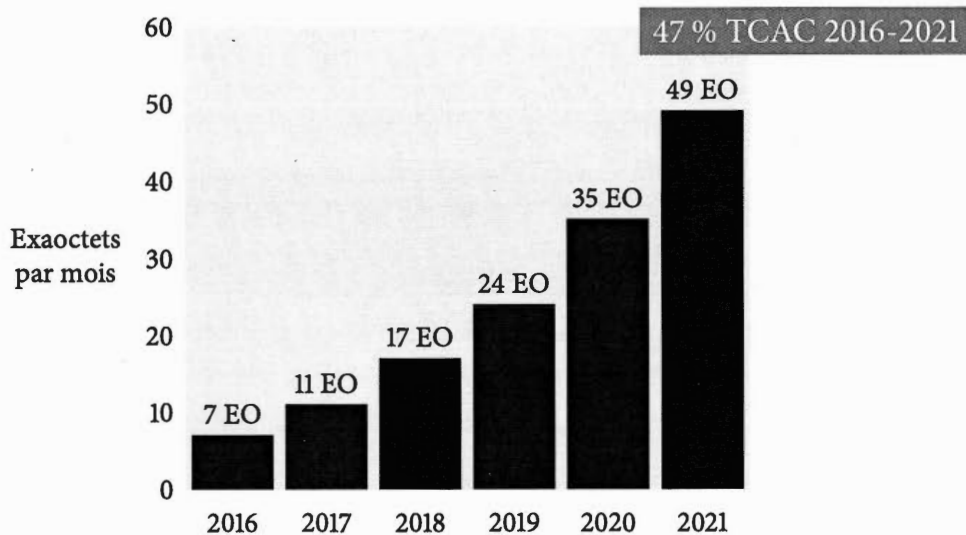


Figure 1.1 Prévisions Cisco pour la croissance du trafic de données mobile (Cisco, 2017).

nentielle du trafic sans fil en Amérique du Nord entre 2007 et 2020. Andrews *et al.* ont surtout montré que, bien que le trafic sans fil croisse et continuera de croître au fil des ans, le trafic de données sans fil augmente très rapidement par rapport aux autres types de trafic. Cette étude est illustrée sur la figure 1.2.

Afin de répondre à cette demande grandissante, plusieurs technologies sans fil ont émergé telles que : a) les communications millimétriques (*millimeter wave*, en anglais), b) les antennes multiples à grande échelle (*massive MIMO*, en anglais) et c) les réseaux denses hétérogènes multi-niveaux ou, en anglais, *dense heterogeneous multi-tier networks* (Quek *et al.*, 2013; Hossain *et al.*, 2013). Ces différentes technologies utilisent plusieurs techniques sans fil afin d'améliorer l'expérience des consommateurs (ou utilis-

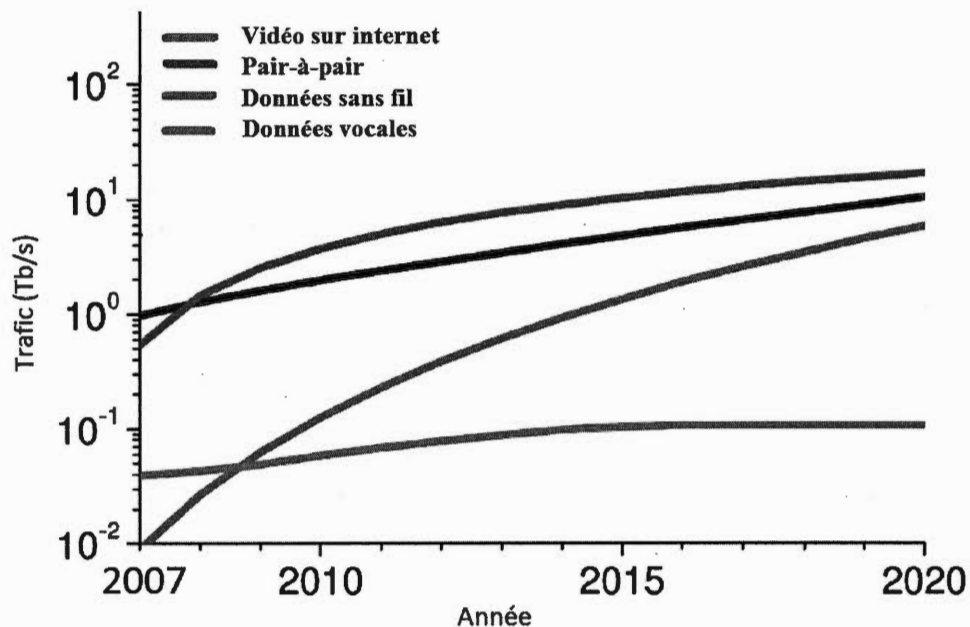


Figure 1.2 Augmentation du trafic sans fil en Amérique du Nord entre 2007 et 2020 (Andrews *et al.*, 2012). L'axe des ordonnées mesure la quantité de trafic en téraoctet par seconde (Tb/s).

teurs) et d'augmenter la capacité des réseaux ¹. Les techniques sans fil fondamentales qui sont utilisées dans ces technologies sont : l'utilisation intelligente du spectre radio (avoir plus de Hertz), l'utilisation d'un nombre élevé d'antennes (avoir plus de bits par seconde par Hertz) et l'augmentation de la densité du réseau (avoir plus de stations de base par unité de surface et par Hertz) (Andrews *et al.*, 2014; Fehske *et al.*, 2014). Ces nouvelles technologies sont considérées comme étant les meilleurs candidats de la prochaine génération des réseaux cellulaires (Andrews *et al.*, 2014). Historiquement, l'augmentation de la densité des réseaux cellulaires a amélioré leur capacité plus que les deux autres techniques (Fehske *et al.*, 2014). Ainsi, dans cette thèse, nous considérons cette technique et nous l'étudions afin de fournir des solutions algorithmiques efficaces qui permettent de

1. Dans cette thèse, nous adoptons la définition de la capacité comme étant le nombre de requêtes (ou utilisateurs) servis.

satisfaire à la demande grandissante des utilisateurs des futurs réseaux cellulaires. L'augmentation de la densité des réseaux cellulaires (ou la densification) peut être réalisée par plusieurs méthodes : soit par une installation distribuée des antennes, soit par l'ajout de nœuds relais ou soit par l'ajout de plus de stations de base de grande taille (autrement dit, les stations de base macro-cellules) (Chandrasekhar *et al.*, 2008). Toutefois, l'installation des macro-cellules s'avère souvent très coûteuse pour les opérateurs des réseaux cellulaires puisqu'elle nécessite une modification importante de l'infrastructure existante.

Un nombre grandissant de travaux de recherche ont poussé les opérateurs des réseaux cellulaires à adopter une nouvelle solution, soit celle du déploiement de petites cellules (*small-cell base stations*, en anglais). Cette solution a été adoptée par les opérateurs grâce à ses avantages puisqu'elle réduit énormément les dépenses en termes des dépenses d'exploitation (ou *operational expenditure* (OPEX) en anglais) et des dépenses d'immobilisation (ou *capital expenditure* (CAPEX) en anglais). En outre, elle offre une grande efficacité spectrale par région, c'est-à-dire, elle permet d'avoir plus de bits par seconde par Hertz par mètre carré (bits/s/Hz/m^2) (Chandrasekhar *et al.*, 2008), puisqu'elle permet de réduire énormément la distance entre les émetteurs et les récepteurs. Ainsi, elle permettra de satisfaire la demande grandissante du trafic de données sans fil (Nguyen et Kountouris, 2017; Chandrasekhar *et al.*, 2008; Andrews *et al.*, 2014). À titre d'exemple, la figure 1.3 illustre l'augmentation de l'efficacité spectrale en fonction de la densité des anciennes générations des réseaux cellulaires comme *long term evolution* (LTE).

Cette thèse se concentre sur les réseaux denses hétérogènes multi-niveaux (simplement dénotés par les réseaux DCNs—*dense cellular networks*). Les réseaux DCNs sont des réseaux cellulaires où plusieurs stations de base coexistent dans la même petite zone géographique. Parmi ces stations de base, nous trouvons notamment les stations de base macro-cellules, les stations de base micro-cellules, les stations de base pico-cellules, et

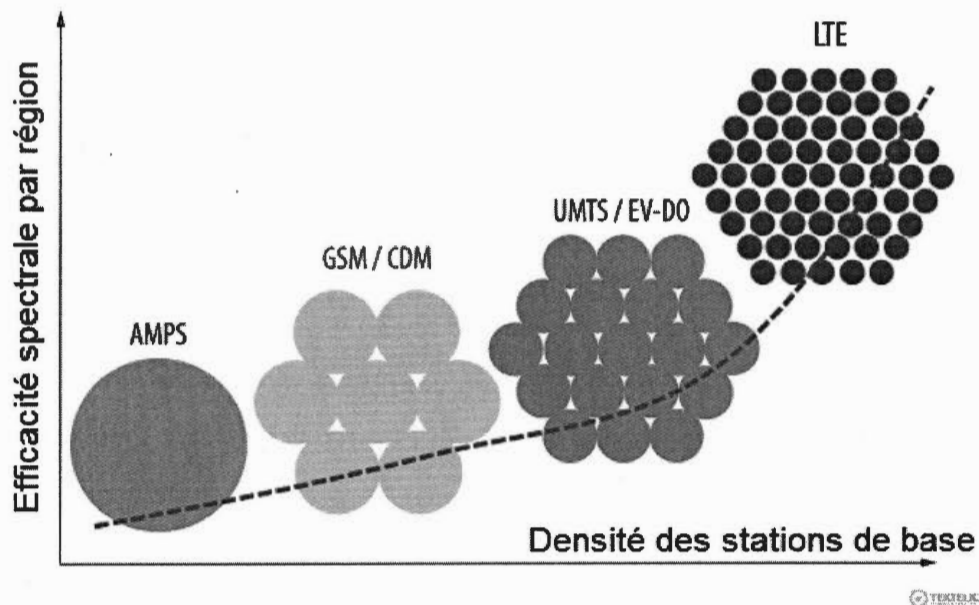


Figure 1.3 Augmentation de l'efficacité spectrale en fonction de la densité des réseaux cellulaires (TEKTELIC Communications, 2014).

les stations de base femto-cellules². Les réseaux DCNs sont considérés comme des réseaux hétérogènes multi-niveaux grâce aux différents types de stations de base qui co-existent ensemble. En effet, chaque type de stations de base forme un niveau dans lequel les stations de base ont pratiquement la même taille, la même puissance de transmission, la même couverture, la même capacité de calcul, le même prix, etc. (Bhushan *et al.*, 2014; Fehske *et al.*, 2014). Les stations de base macro-cellules sont les plus grandes en taille avec une capacité de calcul élevée et une puissance de transmission de l'ordre de 46 dBm. Les stations de base micro-cellules et les stations de base pico-cellules sont petites et leurs capacités de calcul sont faibles par rapport aux stations de base macro-cellules. Elles sont installées, généralement, par les opérateurs dans les rues, les entreprises ou les hôtels. Les stations de base micro-cellules utilisent une puissance de transmission

2. Pour simplifier les notations, nous utilisons, dans la plupart des parties de cette thèse, le terme « petite station de base » pour désigner une station de base femto-cellule, pico-cellule, ou micro-cellule.

de l'ordre de 36 dBm alors que les stations de base pico-cellules utilisent une puissance de transmission de l'ordre de 30 dBm. Finalement, les femto-cellules sont généralement déployées par les utilisateurs et par conséquent sont installées dans les maisons et les bureaux privés. Elles sont regroupées en trois catégories : (i) les femto-cellules à accès ouvert qui peuvent être utilisées par leurs propriétaires ainsi que les non-propriétaires, (ii) les femto-cellules à accès fermé qui peuvent être utilisées seulement par leurs propriétaires et (iii) les femto-cellules à accès hybride qui se caractérisent par un accès à la fois ouvert et fermé. Les femto-cellules utilisent une puissance de transmission de l'ordre de 23 dBm et elles ont une capacité de calcul plus ou moins faible (Quek *et al.*, 2013; Fehske *et al.*, 2014). Les réseaux DCNs sont reconnus tant par les chercheurs académiques que par leurs homologues industriels comme étant les parfaits candidats pour satisfaire aux exigences de la prochaine génération des réseaux cellulaires. En effet, selon des études préliminaires, les réseaux DCNs font partie d'un ensemble de trois technologies qui contribueront à l'augmentation du débit de transmission par un facteur allant jusqu'à 1000 et à un élargissement considérable de la couverture et une amélioration de la qualité d'expériences des utilisateurs (Andrews *et al.*, 2014; Bhushan *et al.*, 2014).

Malgré toutes les promesses apportées par les réseaux DCNs, leur déploiement donne naissance à plusieurs nouveaux défis qui n'existaient pas auparavant. En effet, l'hétérogénéité qui résulte de la cohabitation de plusieurs types de stations de base et la densité de cette nouvelle forme de réseaux aggravent le problème d'interférence en multipliant ses sources et diversifient davantage le nombre des ressources disponibles. Ainsi, l'optimisation de l'allocation des ressources devient un problème multidimensionnel souvent très difficile à résoudre. En outre, le déploiement aléatoire, dynamique et prêt-à-fonctionner (*plug-and-play* en anglais) de différentes stations de base dans ces réseaux ainsi que leurs différentes caractéristiques nécessitent à repenser l'allocation des ressources traditionnelles et de trouver des nouvelles idées et algorithmes efficaces (Andrews *et al.*, 2014; Lopez-Perez *et al.*, 2014). Dans les réseaux DCNs, souvent il existe deux types d'in-

terférence : l'interférence entre le même niveau (*co-tier interference* en anglais) et l'interférence entre deux ou plusieurs niveaux (*cross-tier interference* en anglais). Ces interférences doivent être gérées efficacement. Par conséquent, la maximisation des performances des réseaux DCNs doit passer obligatoirement par une optimisation efficace de l'allocation des ressources dont disposent ces derniers afin de bien gérer l'interférence. Les principales ressources qui existent dans les réseaux DCNs sont l'espace, la puissance (ou l'énergie), la fréquence, et le temps. Cette thèse répond à la problématique générale suivante : comment pouvons-nous gérer l'interférence et optimiser les ressources d'une manière optimale ou presque optimale dans les DCNs ? Autrement dit, comment pouvons-nous concevoir des algorithmes et des protocoles de communications qui sont simples et qui visent à allouer les ressources dont disposent les réseaux DCNs d'une manière optimale ou presque optimale ? Dans ce qui suit, nous discutons les quatre différentes ressources et nous présentons les problématiques spécifiques d'allocations qui concernent chacune d'entre elles.

1.1.1 Les ressources dans les réseaux DCNs

1.1.1.1 La ressource spatiale

La première ressource que nous optimisons dans cette thèse est l'espace. Cette ressource est la ressource la plus importante dans notre travail. L'espace dans ce contexte est défini comme les stations de base elles-mêmes existantes dans les réseaux DCNs. En fait, lorsque différentes stations de base coexistent dans une zone géographique donnée, la question que pose l'utilisateur est : à quelle station de base dois-je me connecter ? On a répondu à cette question dans les anciennes générations des réseaux cellulaires en appliquant la règle de signal le plus fort (appelée en anglais, *maximum signal to interference-plus-noise ratio (max-SINR)*). La règle de max-SINR permet d'associer un utilisateur à la station de base qui fournit le signal le plus fort (mesuré par le SINR). Malheureusement,

cette règle donne des mauvaises performances dans les réseaux DCNs. En effet, si cette règle est appliquée dans un réseau DCN, plusieurs utilisateurs peuvent être sanctionnés (ne seront pas satisfaits) parce qu'ils reçoivent des interférences importantes. En outre, le max-SINR augmente la charge des stations de base macro-cellules puisqu'elles utilisent les puissances de transmission les plus élevées par rapport aux autres stations de base. Pour mieux illustrer la nécessité d'optimiser la ressource spatiale, nous discutons dans ce qui suit certains exemples concrets (Lopez-Perez *et al.*, 2011).

Prenons l'exemple du réseau représenté dans la figure 1.4. Dans cette figure, il existe une station de base macro-cellule (MBS) et quatre petites stations de base (SBS) déployées par un opérateur quelconque d'un réseau DCN. Dans la partie supérieure gauche de la figure 1.4, le signal envoyé de l'utilisateur u_7 vers la MBS dérange beaucoup le signal montant envoyé de l'utilisateur u_6 vers la SBS₃ parce que u_7 est situé dans la zone de couverture de SBS₃. Ce problème serait évité si u_7 était associé au SBS₃. Un autre problème d'interférence survient sur le lien descendant illustré dans la partie inférieure gauche de la figure 1.4. L'utilisateur u_1 connecté à la MBS est dérangé par la SBS₁ parce qu'il est situé à la bordure de la MBS et il est proche de SBS₁. Dans la partie gauche de la figure 1.4, si la MBS, la SBS₁ et la SBS₃ s'associent correctement aux utilisateurs u_1 , u_2 , u_6 et u_7 , ce problème d'interférence pourrait être éliminé et ainsi la capacité du réseau pourrait être améliorée. Prenons maintenant un autre exemple. Dans la figure 1.4, si le nombre d'utilisateurs augmente alors la MBS sera très congestionnée puisque la majorité d'utilisateurs y seront associés³. La solution est de décharger la MBS en associant certains utilisateurs aux SBSs. Cette solution permet à la fois de décharger la MBS, d'améliorer la qualité d'expérience des utilisateurs et d'augmenter la capacité du réseau. En effet, si l'utilisateur u_9 de la partie supérieure droite de la figure 1.4 est associé à la SBS₄ au lieu d'être associé à la MBS, u_8 et u_9 auront une meilleure qualité d'expérience et la charge

3. Dans cette thèse, nous utilisons, de manière interchangeable, les mots « associé », « servi », ou « ordonné ».

de la MBS s'améliorera. Ce problème d'interférence s'aggrave si le nombre de SBSs augmente. Par conséquent, une technique efficace d'association d'utilisateurs est nécessaire afin de résoudre ces problèmes d'interférence et par la suite maximiser les avantages des réseaux DCNs. Nous nous intéressons ainsi à repenser l'optimisation de la ressource spatiale dans les réseaux DCNs afin d'obtenir des solutions d'associations simples en termes de complexité de calcul et efficaces en termes de performances.

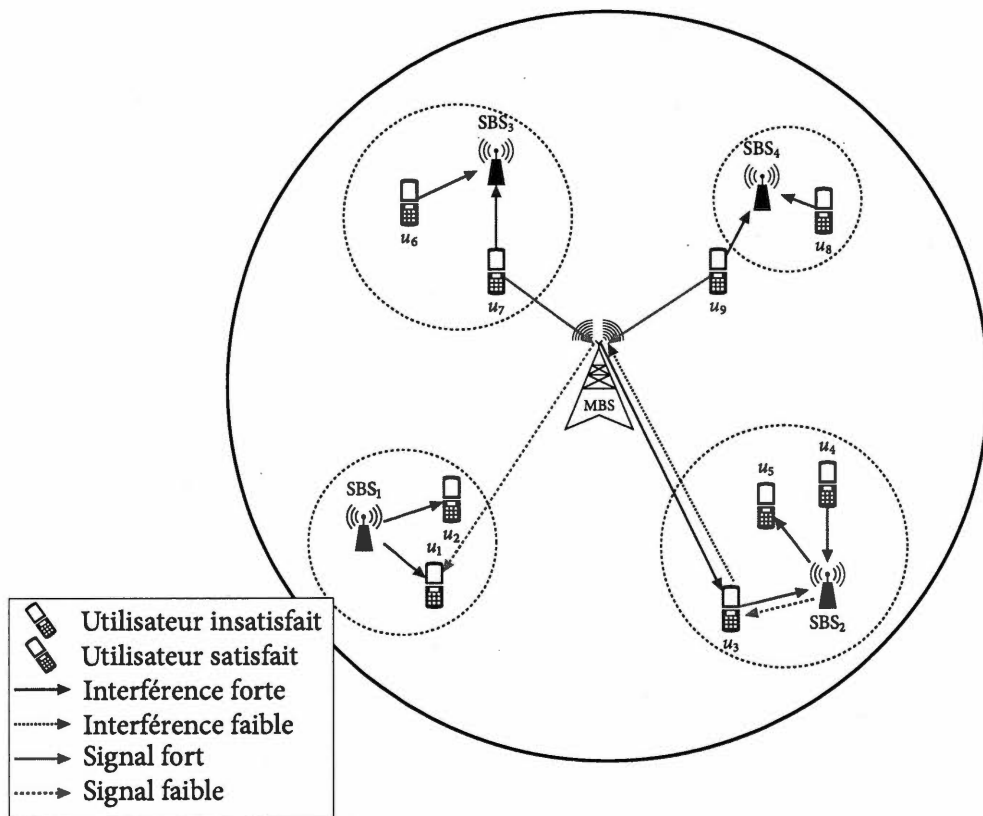


Figure 1.4 Illustration du problème d'interférence et d'association des utilisateurs aux stations de base dans les réseaux DCNs. L'idée de cette figure vient d'une figure similaire publiée dans (Lopez-Perez *et al.*, 2011).

1.1.1.2 La ressource énergétique

Les appareils sans fil et mobiles, utilisés par les utilisateurs des réseaux DCNs, sont généralement équipés de batteries à capacités limitées. Évidemment, plus la durée de vie des batteries des ces appareils est longue, plus la satisfaction des utilisateurs est élevée. En outre, les petites stations de base (p. ex. les femto-cellules) dans les réseaux DCNs sont équipées aussi de batteries à capacités limitées. D'autre part, les stations de base en général consomment énormément d'énergie—entre 60% et 80% de l'énergie totale des réseaux DCNs est consommée par les stations de base (Rao et Fapojuwo, 2014; Feng *et al.*, 2013). Bien que différentes petites stations de base utilisent des puissances de transmission relativement faibles, leur nombre élevé augmente beaucoup la consommation énergétique (Zhuang *et al.*, 2016). Pire encore, les stations de base consomment généralement de l'énergie même avec peu d'activité (Li *et al.*, 2016). Tout ceci implique que la ressource énergétique dans les réseaux DCNs doit être allouée (aux utilisateurs ou aux stations de base) de manière optimale ou, le cas échéant, presque optimale. Dans ce contexte, un nombre important de travaux de recherche a vu le jour afin de (i) minimiser la consommation de l'énergie, ou (ii) maximiser l'utilisation d'énergie verte (Shan *et al.*, 2015; Lee *et al.*, 2017). Parmi les solutions proposées pour maximiser l'utilisation d'énergie verte, nous citons la technologie de récolte d'énergie (*energy harvesting* en anglais). La récolte d'énergie est une technologie qui permet de tirer l'énergie des sources (naturelles) externes telles que solaire, éolienne, thermique, cinétique, etc. (Buzzi *et al.*, 2016a). L'énergie tirée est emmagasinée dans des batteries afin de garantir un fonctionnement autonome. Malgré les avantages apportés par la récolte d'énergie, la nature des sources externes utilisées dans une telle technologie impose souvent deux contraintes importantes (Vaze, 2013) : (i) contraintes d'imprévisibilité, c'est-à-dire que l'énergie récoltée disponible est généralement imprévisible et (ii) contraintes de neutralité, c'est-à-dire que la quantité d'énergie utilisée par les stations de base à tout moment ne peut

pas être supérieure à la quantité d'énergie récoltée disponible. Par conséquent, afin de profiter des avantages des réseaux DCNs, il est obligatoire de bien gérer l'allocation de la ressource énergétique. Dans cette thèse, nous proposons des techniques d'optimisation de la ressource énergétique à travers des algorithmes qui (i) minimisent la consommation énergétique totale en activant seulement les meilleures stations de base ou (ii) maximisent l'utilisation verte de l'énergie en déployant des stations de base avec récolte d'énergie.

1.1.1.3 La ressource fréquentielle

La fréquence est une ressource très importante qui doit être gérée dans le cadre des communications sans fil en général. En effet, le spectre électromagnétique, composé d'intervalles de fréquences, est une ressource naturelle d'une grande importance. Toutefois, cette ressource est très limitée et elle est partagée par plusieurs réseaux. Les générations précédentes des réseaux cellulaires (p. ex. 3G et 4G) utilisent des fréquences dans l'intervalle [600 MHz, 6 GHz]. La prochaine génération des réseaux cellulaires (p. ex. la 5G) utilisera aussi des fréquences dans cet intervalle ainsi que des fréquences extrêmement hautes dans l'intervalle [24 GHz, 86 GHz] (3GPP, 2017). La tâche de partage et de gestion de spectre est souvent assurée par des organismes spécialisés. Par exemple, aux États-Unis, la commission fédérale des communications (FCC) gère le spectre radio des applications non gouvernementales alors que l'administration nationale des télécommunications et de l'information (NTIA) gère le spectre radio des applications gouvernementales. Au Canada, la gestion du spectre radio est assurée par Innovation, sciences et développement économique Canada (ISED) et par le conseil de la radiodiffusion et des télécommunications canadiennes (CRTC). Une fois qu'une partie du spectre radio est attribué, par exemple, aux compagnies de télécommunications, ces dernières doivent l'utiliser d'une manière efficace afin de satisfaire les demandes grandissantes de leurs utilisateurs. Par conséquent, les bandes de fréquence qui constituent cette partie du spectre

doivent être allouées efficacement aux utilisateurs. Dans les réseaux DCNs, nous trouvons généralement deux approches d'allocation de fréquences : l'allocation universelle et l'allocation fractionnelle. La première approche permet aux différentes stations de base de réutiliser la même fréquence, ce qui augmente l'efficacité spectrale du réseau, si l'interférence est bien gérée (Elsawy *et al.*, 2013; Zhuang *et al.*, 2017a). La deuxième approche fractionne les fréquences en des sous-fréquences (ou sous-canaux) et permet aux stations de base (i) de réutiliser la même partie de ces sous-canaux pour les utilisateurs proches (c.-à-d., appliquer une allocation universelle) et (ii) d'allouer le reste des sous-canaux pour les utilisateurs lointains en utilisant un motif de réutilisation précis (c.-à-d., certains utilisateurs lointains utilisent les mêmes sous-canaux, d'autres utilisent d'autres sous-canaux) (Ghaffar et Knopp, 2010). Afin de maximiser l'efficacité spectrale des réseaux DCNs, nous proposons dans une partie de cette thèse des algorithmes d'allocation de ressources qui permettent de bien gérer les sous-canaux entre les utilisateurs.

1.1.1.4 La ressource temporelle

Une autre ressource également importante est le temps. La gestion de la ressource temporelle est connue souvent sous le nom d'ordonnancement (Pinedo, 2016). L'objectif de l'ordonnancement est de trouver les moments optimaux d'exécution des tâches afin de satisfaire des performances requises (p. ex. délai minimal, capacité maximale, etc.). Dans un réseau DCN, lorsque les utilisateurs sont associés à une station de base pour obtenir un certain service, cette station de base partage ses ressources entre eux. Parmi ces ressources partagées, nous trouvons évidemment la ressource temporelle. Pour que la station de base puisse partager cette ressource, elle la divise souvent en des intervalles de temps (ou *slots* en anglais)⁴. Cette répartition est souvent appelée en anglais *time division multiple access* (TDMA). Pendant chaque intervalle de temps et selon la condition

4. Dans cette thèse nous utilisons le terme « *slot* » à la place « d'intervalle de temps ».

du médium physique sans fil, la puissance ainsi que la fréquence de la station de base seront allouées aux différents utilisateurs associés. Mais, la question qui se pose afin de maximiser les performances des réseaux DCNs est la suivante : à quel intervalle de temps la station de base doit ordonnancer quel utilisateur, avec quelle puissance et sur quelle fréquence ? Afin de répondre à cette question, une méthode d'accès multiple efficace est nécessaire. Par conséquent, cette thèse propose des algorithmes efficaces qui permettent d'allouer la ressource temporelle aux différents utilisateurs afin de maximiser la capacité des réseaux DCNs.

Malgré le nombre grandissant de travaux de recherche traitant l'allocation des ressources dans les réseaux cellulaires en général et dans les réseaux DCNs en particulier, plusieurs problèmes restent encore ouverts. En effet, comme mentionné précédemment, les différentes caractéristiques des réseaux DCNs nécessitent des algorithmes d'allocation des ressources spécialement adaptés. Premièrement, la complexité computationnelle (ou l'étude des classes de complexité P, NP-difficile, NP-complet, etc.)⁵ des différents problèmes d'allocation des ressources dans les réseaux DCNs est souvent ignorée ou non-rigoureusement démontrée. Deuxièmement, les algorithmes d'allocation des ressources proposés sont généralement des algorithmes heuristiques qui ne garantissent aucune performance dans les pires des cas. Autrement dit, il n'existe aucune garantie théorique qui permette de majorer (ou minorer) les performances de ces algorithmes dans tous les cas. Finalement, la majorité des algorithmes proposés sont généralement des algorithmes centralisés hors lignes (*offline*), ce qui veut dire qu'ils sont implémentés dans un point central (souvent la station de base macro-cellule) et qui nécessitent toutes les entrées du problème pour être fonctionnels, c.-à-d., les utilisateurs, l'arrivée de l'énergie, etc. doivent être disponibles avant l'exécution de ces algorithmes. Évidemment, le point central constitue souvent un goulot d'étranglement et impose un point de dé-

5. Dans cette thèse, nous utilisons P (resp. NP) pour désigner la classe de complexité polynomiale (resp. non-déterministe polynomiale). Dans la section 1.3.1, nous donnons les définitions nécessaires.

faillance unique. En outre, la nécessité d'avoir toutes les entrées du problème doit être relaxée parce qu'en pratique, dans un DCN, l'entrée (les utilisateurs, l'arrivée de l'énergie, etc.) n'est pas toujours disponible auparavant. D'où la nécessité de concevoir des algorithmes distribués et en ligne (ou *online* en anglais)⁶. Cette thèse comble ces lacunes par :

- une étude rigoureuse de la complexité computationnelle du problème d'allocation des ressources dans les réseaux DCNs en analysant sa NP-difficulté;
- une conception des algorithmes *offlines* centralisés d'approximation qui se caractérisent par leur garantie théorique sur les valeurs des solutions obtenues;
- une conception des algorithmes distribués d'apprentissage ainsi que des algorithmes *onlines* compétitifs qui se caractérisent par leur exécution autonome et leur simplicité;
- une analyse asymptotique de la complexité algorithmique des différents algorithmes proposés; et
- une simulation des différents résultats obtenus.

Dans la section suivante, nous discutons les objectifs de cette thèse en détails.

1.2 Objectifs

1.2.1 L'objectif général

Dans cette thèse, nous considérons le problème d'allocation des ressources (spatiales, énergétiques, fréquentielles, et temporelles) dans les réseaux DCNs. L'objectif général de cette thèse est d'étudier la complexité computationnelle du problème d'allocation des ressources et de le résoudre en proposant des algorithmes centralisés et distribués simples

6. Dans le reste de la thèse, nous utilisons les termes « *online* » et « *offline* » au lieu des termes « en ligne » et « hors ligne », respectivement.

et efficaces. Les solutions proposées maximisent la capacité des réseaux DCNs en garantissant une utilisation optimale (ou presque optimale) des différentes ressources.

1.2.2 Les objectifs spécifiques

Le problème d'allocation des ressources est un problème d'optimisation combinatoire par nature, c.-à-d. la solution optimale doit être sélectionnée parmi un ensemble discret de solutions candidates. Souvent, les problèmes d'optimisation combinatoires sont « difficiles » à résoudre. Un objectif important de cette thèse consiste donc à caractériser mathématiquement la « difficulté » du problème d'allocation des ressources. Autrement dit, nous analysons la complexité computationnelle de ce problème en étudiant sa NP-difficulté. De manière informelle, un problème est dit NP-difficile s'il est presque impossible de le résoudre d'une manière optimale dans un temps raisonnable. C'est-à-dire que, sous l'hypothèse de $P \neq NP$, il est impossible de trouver un algorithme efficace qui résout un problème NP-difficile, où nous suivons la convention selon laquelle un algorithme efficace est celui qui s'exécute dans un temps limité par un polynôme dans la taille du problème (nous appelons un tel algorithme un algorithme polynomial). Alors, qu'est-ce que nous pouvons faire dans ce cas ?

Ainsi, cette thèse répond à cette question. En effet, après avoir étudié la NP-difficulté du problème d'allocation des ressources, nous devons le résoudre d'une manière efficace. Un vieux slogan d'ingénierie dit : « *Fast. Cheap. Reliable. Choose Two* ». Donc, malheureusement, sous l'hypothèse largement répandue de $P \neq NP$, nous ne pouvons pas trouver un algorithme qui résout un problème NP-difficile et qui garantit à la fois : (i) une solution optimale (ii) dans un temps polynomial (iii) pour toutes les instances du problème. Il faut alors relaxer au moins une de ces trois exigences afin de résoudre un tel problème. D'une manière générale, une première approche relaxe l'exigence de « pour toutes les instances du problème » et donc cherche à résoudre d'une manière optimale des cas particu-

liers. Une deuxième approche plus commune est de relaxer l'exigence de « dans un temps polynomial ». Cette approche cherche donc à résoudre un tel problème en trouvant des algorithmes optimaux qui ne se limitent pas à des temps polynomiaux. Dernièrement, l'approche la plus commune est de relaxer l'exigence de « une solution optimale ». Cette approche cherche à résoudre un problème NP-difficile en proposant un algorithme qui retourne, pour toutes les instances, une « bonne » solution dans un temps polynomial. Cette approche est particulièrement importante dans la pratique parce qu'elle permet de trouver des algorithmes qui s'exécutent en quelques secondes ou même moins. Elle a beaucoup été utilisée dans la littérature en proposant des algorithmes heuristiques ou méta-heuristiques. Ces algorithmes se caractérisent principalement par le fait qu'ils produisent des « bonnes » solutions. Toutefois, une question importante qui se pose maintenant est la suivante : qu'est-ce qu'une « bonne » solution ? Sauf indication contraire, dans cette thèse nous adoptons la définition d'une « bonne » solution comme celle qui offre une garantie théorique sur sa valeur. Malheureusement, pour la majorité des algorithmes proposés dans la littérature, même s'ils fournissent des solutions efficaces, ces dernières n'ont aucune garantie théorique sur leurs valeurs dans tous les cas. Un de nos objectifs dans cette thèse consiste donc à proposer des algorithmes polynomiaux d'allocation des ressources qui fournissent théoriquement des bonnes solutions. Autrement dit, nous proposons, lorsque c'est possible, des *algorithmes d'approximation*. Par exemple, si l'objectif est de maximiser la capacité d'un réseau DCN et que la solution optimale est de X bits/s, alors une bonne solution est celle qui donne une capacité qui n'est pas pire de αX (avec $\alpha \geq 1$).

Les algorithmes d'approximation que nous proposons dans cette thèse sont considérés comme des algorithmes centralisés *offlines*. Afin de réduire la dépendance à des points centraux qui constituent souvent des points de défaillance uniques, un objectif de cette thèse sera donc de concevoir des algorithmes *offlines* distribués ou d'apprentissage et des algorithmes *online*s compétitifs. D'une part, les algorithmes d'apprentissage per-

mettent d'avoir des solutions d'allocation des ressources efficaces et autonomes. D'autre part, les algorithmes *onlines* compétitifs peuvent produire des bonnes solutions sans connaître la totalité de l'entrée du problème, c'est-à-dire, ces algorithmes ne reçoivent pas leur entrée d'un seul coup, mais comme un flux de données, et doivent prendre des décisions irrévocables au fur et à mesure.

1.2.3 Résumés des objectifs spécifiques

Les objectifs spécifiques de cette thèse peuvent être résumés comme suit. Nous :

- modélisons les problèmes d'allocation des ressources mathématiquement d'une manière conjointe et les résoudre analytiquement (si possible). Cet objectif permet de fournir les solutions optimales en les modélisant comme des programmes mathématiques et en les résolvant, par exemple, par les algorithmes de séparation et évaluation (*Brand and Bound* (BnB) en anglais). Ces algorithmes seront considérés comme des références de comparaison pour les performances;
- étudions la classe de complexité (les classes NP-difficile, NP-complet, P, NP, etc.) des problèmes d'allocation des ressources. Cet objectif permet de savoir quelle méthode est la plus adéquate pour résoudre ce genre de problèmes;
- concevons des algorithmes *offlines* d'approximation simples en termes de complexité de calcul et efficaces en termes de performances qui garantissent théoriquement les qualités des solutions proposées;
- concevons des algorithmes *offlines* distribués et autonomes qui se caractérisent par leur indépendance aux points centraux;
- concevons des algorithmes *onlines* compétitifs qui sont adaptatifs et qui se caractérisent par leur garantie de performance;
- analysons la complexité algorithmique asymptotique en termes de temps d'exécution de chaque algorithme conçu; et finalement

- évaluer les performances des différentes solutions obtenues en les comparant avec les solutions existantes ainsi que les solutions optimales.

1.3 Méthodologie générale

Dans cette thèse, nous étudions des problèmes d'allocation des ressources dans les réseaux DCNs en proposant des algorithmes simples et efficaces. Nous suivons la méthodologie de recherche suivante. D'abord, nous réalisons une recherche bibliographique pertinente de la littérature afin de spécifier les lacunes qui existent dans le cadre de l'allocation des ressources dans les réseaux DCNs. En particulier, une étude de l'état de l'art de l'allocation des ressources dans les réseaux cellulaires et les réseaux DCNs est effectuée. Cette étude bibliographique nous permet de mieux comprendre la problématique ainsi que de savoir les points faibles des solutions existantes. Après l'étude bibliographique, nous étudions différents modèles de système des réseaux DCNs. Pour chaque modèle, nous présentons les hypothèses utilisées et nous définissons la problématique. En effet, cette modélisation caractérise le canal de communication sans fil ainsi que les signaux à transmettre en prenant en considération l'affaiblissement de propagation, l'évanouissement des trajets, la dynamique de trafic, etc. Ensuite, nous formulons les problèmes comme étant des problèmes d'optimisation mathématique en précisant les fonctions objectif et les contraintes correspondantes. Parmi les exemples des fonctions objectif, nous utilisons la capacité du réseau et l'efficacité énergétique. Les exemples des contraintes sont les limites qui caractérisent le réseau telles que la puissance limitée ou la charge de la station de base. Cette modélisation utilise les techniques de programmation linéaire et non-linéaire en nombre entiers. Elle permet d'obtenir les solutions optimales en utilisant par exemple les algorithmes de séparation et d'évaluation ou la génération de colonnes. Cette étape donne, souvent, des solutions complexes qui sont utilisées pour des fins de comparaison. Dans cette étape, nous utilisons des outils mathématiques combinatoires et des outils de résolution des programmes mathématiques tels que CPLEX, Gurobi ou

GLPK. (IBM, 2010; Gurobi Optimization Inc., 2016; GLPK, 2006) qui seront implémentés dans des outils de résolution comme PuLP (Mitchell *et al.*, 2011) avec Python ou JuMP (Lubin et Dunning, 2015) avec Julia. Ensuite, nous formulons les problèmes considérés comme étant des problèmes de décision afin d'étudier leur complexité computationnelle (c.-à-d. NP-difficulté ou NP-complétude). Cette étude utilise les techniques de réduction algorithmiques comme la réduction par restriction ou par remplacement local. L'étape suivante consiste à développer des algorithmes *offlines* d'approximation ou *offline* heuristiques, selon le contexte. Cette étape utilise des techniques de conception algorithmiques telles que la méthode diviser pour régner, la méthode gloutonne ou des méthodes algorithmiques similaires. Puisque ces algorithmes sont considérés comme des algorithmes centralisés, nous développons dans l'étape suivante des algorithmes distribués et d'apprentissage. Nous utilisons les techniques de la théorie des jeux ainsi que celles de la théorie d'intelligence artificielle et de probabilité afin de produire des solutions plus robustes aux changements futurs. Jusqu'à maintenant, nous avons utilisé dans notre méthodologie des algorithmes dits *offline*; ceux qui s'exécutent en connaissant l'entrée en totalité. Toutefois, dans les réseaux DCNs, il existe des cas où l'entrée n'est pas disponible en totalité, comme par exemple, l'arrivée d'utilisateurs. Par conséquent, dans cette étape, nous concevons des algorithmes *onlines* qui sont compétitifs, autrement dit des algorithmes *onlines* d'approximation. Dans cette étape, nous utilisons les techniques algorithmiques citées précédemment comme par exemple la méthode gloutonne ou l'analyse compétitive. Enfin, nous étudions la complexité algorithmique et le temps d'exécution de chaque algorithme conçu en utilisant des analyses mathématiques asymptotiques. La dernière étape est l'évaluation des performances par simulations. La validation des résultats est basée sur plusieurs comparaisons avec les travaux connexes existants dans la littérature⁷ ainsi qu'avec les solutions optimales. Dans cette étape, des simulations infor-

7. Dans certains cas, il faut adapter les solutions proposées dans la littérature à nos problèmes afin de pouvoir les utiliser comme des références de comparaison.

matiques, telle que la simulation Monte Carlo ⁸, seront fournies en utilisant un langage de programmation, soit Python ou Julia. Notre méthodologie est résumée sur la figure 1.5.

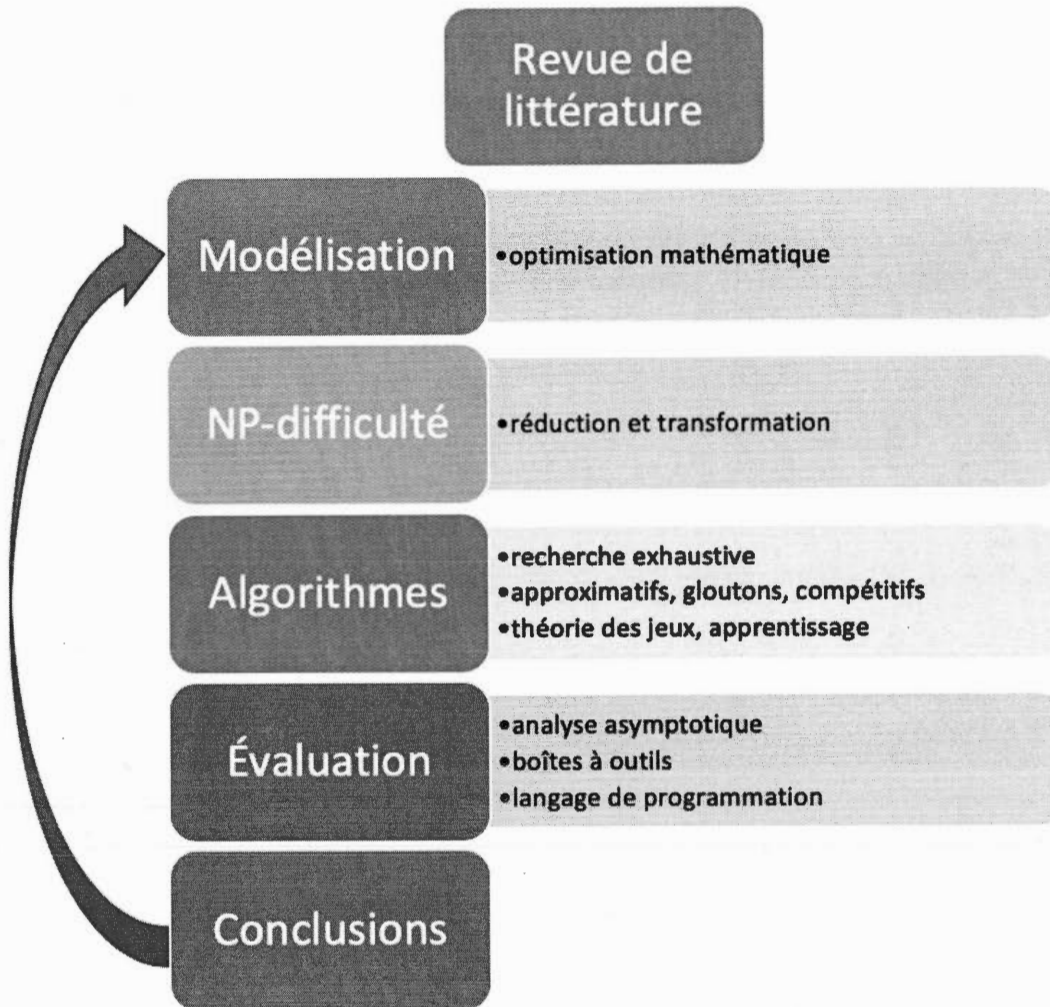


Figure 1.5 La méthodologie de la thèse.

8. Dans cette thèse, la méthode de Monte Carlo est utilisée parce que les réseaux étudiés ont plusieurs paramètres aléatoires et il est difficile d'utiliser des modèles mathématiques exacts.

1.3.1 Définitions

Dans cette partie, nous définissons quelques aspects liés à notre travail.

Définition 1.3.1 ((Informelle) P, NP, NP-difficile, et NP-complet (Arora et Barak, 2009)) :

Les classes de complexité sont les suivantes :

- La classe de complexité P contient tous les problèmes de décisions qui peuvent être résolus en temps polynomial par une machine de Turing déterministe.
- La classe de complexité NP contient tous les problèmes de décisions qui peuvent être résolus en temps polynomial par une machine de Turing non déterministe.
- Un problème est NP-difficile si tous les problèmes de la classe NP se réduisent polynomialement à celui-ci.
- Un problème est NP-complet s'il est un problème de décision et NP-difficile et appartient à NP.

Définition 1.3.2 (Un algorithme α -approximatif (Williamson et Shmoys, 2011)) :

Un algorithme ALG pour un problème de maximisation est dit une α -**approximation** si

$$A \leq O \leq \alpha A, \quad (1.1)$$

où A est la valeur de la solution retournée par ALG et O est la valeur d'une solution optimale. L'inégalité (1.1) doit être satisfaite peu importe les entrées.

Lorsque le problème en est un de minimisation, (1.1) se modifie à l'inégalité suivante :

$$A \leq \alpha O \leq A. \quad (1.2)$$

De même, cette inégalité doit être satisfaite quelque soit I.

Pour les deux inégalités, le facteur d'approximation x est supérieur ou égal à 1; plus il est proche de 1, meilleure est l'approximation.

Définition 1.3.3 (Un algorithme x -compétitif (Borodin et El-Yaniv, 1998)) :

Un algorithme *online* ALG pour un problème de maximisation est x -**compétitif** s'il existe une constante c telle que pour toutes les instances I de P ,

$$o(I) \leq xA(I) + c, \quad (1.3)$$

où $A(I)$ est la valeur retournée par ALG étant donnée I et $o(I)$ est la valeur retournée par un algorithme optimal *offline* OPT. ALG est **strictement** x -**compétitif** s'il est x -compétitif et $c \leq 0$.

De la même manière, le facteur de compétition x est supérieur ou égal à 1; plus il est proche de 1, meilleure est la compétition.

Autrement dit, un algorithme est x -compétitif s'il est x -approximation et *online*. Souvent dans cette thèse, nous nous intéressons à concevoir des algorithmes polynomiaux, donc, sauf indication contraire, un algorithme x -approximation ou x -compétitif est un algorithme polynomial.

1.4 Contributions de la thèse

Ce projet de thèse vise à contribuer à l'avancement des connaissances sur un sujet relativement peu exploré, soit celui de l'allocation des ressources dans les réseaux DCNs. Ce type de réseau constitue sans doute la clé de voûte de la prochaine génération des réseaux cellulaires (Andrews *et al.*, 2014). Par conséquent, nous souhaitons que cette thèse ait une incidence sur : le développement des algorithmes qui permettent d'apporter des solutions théoriques appropriées. Ces résultats sont, au meilleur de nos connais-

sances, à la fois nouveaux et uniques en termes d'efficacité et de praticabilité. Par conséquent, ils constituent des innovations technologiques importantes. Cette thèse fournit une meilleure compréhension du problème d'allocation des ressources et offre des directives pour le résoudre dans les réseaux DCNs de la prochaine génération.

Les différentes contributions réalisées dans ce doctorat sont discutées dans le reste de cette section. Chaque contribution fait l'objet d'une sous-section dans laquelle nous résumons le travail effectué ainsi que les publications scientifiques produites.

1.4.1 Contribution 1 : association centralisée et distribuée des utilisateurs sous les contraintes de SINR

Dans une première partie de ce doctorat, nous étudions le problème d'association des utilisateurs aux stations de base en considérant plusieurs intervalles de temps (*slots*). L'objectif de ce problème est de maximiser le nombre d'utilisateurs servis sur l'ensemble des *slots* sous les contraintes de qualité de service requises par les utilisateurs qui sont représentées par les rapports signal sur interférence-plus-bruit (SINR) (Mlika *et al.*, 2018c; Mlika *et al.*, 2017b). Ce problème est noté par *MSA* pour *multi-slot association*. Nous formulons *MSA* comme un programme linéaire en nombres entiers. Après, nous montrons qu'il est NP-difficile en réduisant à lui, en temps polynomial, le problème d'activation des liens (Halldórsson et Mitra, 2011). Ensuite, nous simplifions le problème en considérant un seul *slot* et nous montrons quand-même que ce problème simplifié (noté par *SSA* pour *single slot association*) reste NP-difficile. Cependant, nous montrons qu'en résolvant *SSA*, nous pouvons produire une solution approximative au problème *MSA* qui est démontrée d'être très proche (elle est au moins égale à $1 - 1/e$ fois la solution optimale) de la solution optimale de *MSA*. Autrement dit, nous démontrons l'existence d'une borne supérieure étroite sur la valeur de la solution optimale. Finalement, nous présentons quelques simulations Monte-Carlo pour illustrer l'efficacité de la méthode proposée

ainsi que pour montrer l'étroitesse de la borne supérieure proposée. Dans une deuxième partie de cette contribution, nous proposons des algorithmes distribués qui résolvent MSA sans la nécessité d'un point central. Pour ce faire, nous utilisons la théorie des jeux ainsi que l'apprentissage automatique. Nous proposons un jeu non coopératif et nous étudions son équilibre de Nash. Nous finissons par l'évaluation de performances des algorithmes distribués via des simulations Monte Carlo.

Cette contribution a été publiée dans les articles de revue *IEEE Communications Letters* 2018 (Mlika *et al.*, 2018c) et *Elsevier Computer Communications* 2017 (Mlika *et al.*, 2017a). Des résultats préliminaires de ce travail ont été présentés à la conférence *IEEE International Conference on Communications*, Paris, France, 2017 (Mlika *et al.*, 2017b).

1.4.2 Contribution 2 : activation des stations de base et association des utilisateurs pour une meilleure efficacité énergétique

Dans une deuxième partie de ce doctorat, nous étudions le problème de minimisation de la consommation énergétique dans les réseaux DCNs (Mlika *et al.*, 2018a; Mlika *et al.*, 2017c). Spécifiquement, nous considérons le problème de minimisation du nombre de stations de base à activer sous les contraintes d'associer tous les utilisateurs avec une qualité de service représentée par un débit de transmission minimal. Nous formulons le problème comme un programme linéaire en nombres entiers. Après, nous étudions ce problème dans deux cas de qualité de service : haute exigence de qualité de service et faible exigence de qualité de service. Dans les deux cas, nous montrons que le problème est NP-complet. Dans le cas de haute exigence de qualité de service, nous montrons que le problème peut être réduit à un problème d'activation de stations de base seulement (c.-à-d., l'association des utilisateurs devient triviale). Ensuite, nous proposons un algorithme optimal basé sur la recherche exhaustive et un autre algorithme heuristique polynomial simple. Dans le cas de faible exigence de qualité de service, nous montrons

que l'activation des stations de base et l'association des utilisateurs ne peuvent pas être découplés. Afin de pouvoir résoudre ce problème, nous considérons le pire cas d'interférence en supposant que toutes les stations de base interfèrent les unes avec les autres. Ensuite, nous proposons un algorithme d'approximation efficace qui permet de : (i) minimiser le nombre de stations de base à activer en garantissant un facteur d'approximation logarithmique et ceci lorsque le problème admet au moins une solution faisable, et (ii) maximiser le nombre d'utilisateurs associés en garantissant un facteur d'approximation constant et ceci lorsque le problème n'est pas faisable. Finalement, nous présentons des simulations numériques afin d'illustrer les performances des algorithmes proposés et de supporter nos résultats théoriques.

Cette contribution a été publiée dans l'article de revue *IEEE Transactions on Wireless Communications* 2018 (Mlika *et al.*, 2018a). Des résultats préliminaires de ce travail ont été présentés à la conférence *IEEE Global Communications Conference*, Singapore, 2018 (Mlika *et al.*, 2017c).

1.4.3 Contribution 3 : ordonnancement, association des utilisateurs et allocation de fréquences sans et avec la récolte d'énergie dans les scénarios *offline* et *online*

Dans une troisième partie de ce doctorat, nous étudions le problème conjoint d'ordonnancement, d'association des utilisateurs aux stations de base et d'allocation des fréquences dans les réseaux DCNs. Ce problème est résolu sous les contraintes de qualité de service représentée par des délais contraignants sans ou avec la technologie de la récolte d'énergie (Mlika *et al.*, 2017d; Mlika *et al.*, 2018b; Mlika *et al.*, 2018d). D'abord, nous formulons le problème comme un programme linéaire en nombres entiers. Puis, nous l'étudions dans le cas d'une seule fréquence avec des stations de base qui ont suffisamment d'énergie (c.-à-d., le cas sans récolte d'énergie). Nous démontrons que ce problème est NP-difficile lorsque les requêtes des utilisateurs ont des temps d'arrivée différents

mais polynomialement soluble sinon. Ensuite, nous concevons un algorithme optimal ainsi qu'un algorithme d'approximation simple et efficace qui garantit un facteur d'approximation constant par rapport à la valeur de la solution optimale. D'autre part, nous traitons le problème général en prenant en considération l'ordonnancement, l'association des utilisateurs ainsi que l'allocation des fréquences avec la technologie de la récolte d'énergie. Nous étudions la NP-difficulté de ce problème dans plusieurs cas et développons des algorithmes d'approximation, heuristiques et optimaux polynomiaux selon le cas. Ce problème est aussi résolu dans un scénario *online* (Mlika *et al.*, 2019), c.-à-d. lorsque l'entrée du problème n'est pas disponible d'un seul coup mais arrive comme un flux de données au fur et à mesure. Nous utilisons l'analyse compétitive afin de proposer des algorithmes *online*s compétitifs (autrement dit, des algorithmes *online*s d'approximation). Nous proposons aussi un algorithme d'apprentissage incrémental qui permet d'obtenir un regret minimal⁹. En fin, nous présentons des simulations Monte Carlo et nous illustrons les performances des algorithmes proposés en les comparant avec des algorithmes de référence de la littérature ainsi qu'avec des algorithmes optimaux.

Cette contribution est partiellement publiée dans l'article de revue *IEEE Communications Letters* (Mlika *et al.*, 2017d). Une partie de cette contribution a été soumise pour évaluation à la revue *IEEE Transactions on Communications* 2018 (Mlika *et al.*, 2018b) et une autre partie soumise à la conférence *ACM Int. Sympos. on Mobile Ad Hoc Netw. and Comput. (ACM MOBIHOC)* 2019. Enfin, une troisième partie a été publiée dans les actes de conférence et présentée à la conférence *IEEE Global Conference on Communications*, Abu Dhabi, Emirates, 2018 (Mlika *et al.*, 2018d).

9. Le regret est défini au niveau du chapitre 4. Informellement, c'est la différence cumulative des gains obtenus par deux algorithmes.

1.5 Organisation de la thèse

Nous présentons, à partir du chapitre suivant, les différentes contributions de notre travail. Chaque chapitre suivant contient : (i) un résumé de la contribution, (ii) une introduction qui contient un rappel aux motivations et problématiques correspondantes, (iii) une revue de littérature détaillée qui analyse les différents travaux connexes et leurs relations avec les notres, (iv) le modèle de système considéré, la formulation du problème ainsi que les solutions proposées et (v) les résultats de simulations. Le reste de la thèse est organisé comme suit. Le chapitre 2 présente notre première contribution, soit celle d'association centralisée et distribuée des utilisateurs sous les contraintes de SINR. Le chapitre 3 présente notre deuxième contribution qui concerne le problème d'activation des stations de base et association des utilisateurs pour une meilleure efficacité énergétique. Ensuite, le chapitre 4 discute de notre contribution trois, soit celle d'ordonnement, association des utilisateurs et allocation de fréquences sans et avec la récolte d'énergie dans les scénarios *offline* et *online*. Finalement, dans le chapitre 5, nous présentons les conclusions et les travaux futurs ainsi que les contributions à l'avancement des connaissances de cette thèse de doctorat.

CHAPITRE II

ASSOCIATION CENTRALISÉE ET DISTRIBUÉE DES UTILISATEURS SOUS LES CONTRAINTES DE SINR

Dans ce chapitre nous présentons le problème d'association des utilisateurs aux stations de base sur plusieurs *slots* dans un réseau DCN sous les contraintes de qualité de service représentées par les rapports signal-sur-interférence-plus-bruit (ou simplement SINR).

Ce travail a été publié dans les articles suivants : (i) *IEEE Communications Letters* 2018 (Mlika *et al.*, 2018c), (ii) *Elsevier Computer Communications* 2017 (Mlika *et al.*, 2017a), et (iii) *IEEE International Conference on Communications*, Paris, France, 2017 (Mlika *et al.*, 2017b). Dans ce chapitre et pour fins de simplicité, nous omettons de présenter les résultats préliminaires publiés dans (Mlika *et al.*, 2017b).

2.1 Résumé

Ce chapitre étudie le problème d'association des utilisateurs aux stations de base sur plusieurs *slots*, nommé *multi-slot association* (MSA). MSA consiste à maximiser le nombre d'utilisateurs associés aux stations de base pendant un certain nombre de *slots*. Cette association doit se faire en garantissant les exigences en qualité de service des utilisateurs associés. Autrement dit, chaque utilisateur associé doit respecter un SINR mi-

nimum. Pour résoudre MSA, nous commençons par étudier sa version *uni-slot*, appelée *single-slot association* (SSA). Ensuite, nous démontrons que SSA et MSA sont NP-difficiles en utilisant une réduction polynomiale à partir du problème d'activation des liens (Halldórsson et Mitra, 2011). Afin de résoudre MSA, nous proposons un algorithme centralisé glouton qui résout SSA dans chaque *slot* et trouve une solution approximative pour MSA. En effet, nous démontrons que la valeur de la solution retournée par cet algorithme atteint une borne supérieure étroite sur la valeur de la solution optimale. L'algorithme glouton proposé est centralisé, et donc, implémenté au niveau d'un point central. Afin de réduire la dépendance à ce point central qui constitue souvent un goulot d'étranglement, nous modélisons MSA en utilisant la théorie des jeux. Ainsi, nous proposons un algorithme d'apprentissage totalement distribué. Enfin, nous présentons les résultats de simulations afin d'illustrer les performances des algorithmes proposés.

2.2 Introduction

Les réseaux DCNs deviennent de plus en plus populaires car ils sont en mesure d'augmenter la capacité des réseaux sans fil en introduisant différentes petites stations de base qui se caractérisent par leur faible puissance et leur petite portée (Andrews *et al.*, 2014). En déployant différentes stations de base dans une zone géographique donnée, l'efficacité spectrale par unité de surface (bits/s/Hz/m^2) peut être augmentée. Cette augmentation est due principalement à la réutilisation agressive des fréquences et au déploiement dense des stations de base (Hanly *et al.*, 2015). Cependant, cette réutilisation de fréquence et ce déploiement engendrent des problèmes d'interférence sévères. Par conséquent, pour améliorer les performances des réseaux DCNs, des nouvelles techniques de gestion d'interférence sont nécessaires (Andrews *et al.*, 2014).

L'association des utilisateurs aux stations de base est l'une des meilleures solutions pour gérer l'interférence dans les réseaux DCNs. Dans les réseaux cellulaires tradition-

nels, l'association des utilisateurs se fait généralement en se basant sur l'algorithme de rapport signal-sur-interférence-plus-bruit maximum (ou max-SINR) (Ye *et al.*, 2016b). Cet algorithme donne des mauvaises performances dans les réseaux DCNs (Ye *et al.*, 2016b). En général, les travaux de recherche qui proposent des algorithmes d'association des utilisateurs (i) utilisent le pire cas d'interférence en supposant que tous les utilisateurs ou toutes les stations de base interfèrent entre eux ou entre elles, respectivement, (ii) utilisent des hypothèses d'interférence simplifiées en modélisant l'interférence par la théorie des graphes et non pas par les contraintes SINR, (iii) ne fournissent pas des analyses théoriques du problème (par exemple, NP-difficulté ou borne supérieure), ou (iv) ne proposent pas des algorithmes totalement distribués. Dans ce chapitre, nous résolvons le problème d'association des utilisateurs dans les réseaux DCNs en utilisant un modèle d'interférence réaliste, c.-à-d. un modèle qui se base sur le rapport SINR où chaque utilisateur (resp. station de base) n'est actif(ve) que si au moins une station de base (resp. utilisateur) y est associé. Autrement dit, l'interférence n'est pas surestimée. Ce problème est nommé *multi-slot association* (MSA). L'objectif de MSA est de maximiser le nombre d'utilisateurs associés pendant un ensemble de *slots* de telle sorte que les exigences en termes de SINR des utilisateurs associés soient respectées. Nous étudions MSA et sa version *uni-slot* (*single-slot association* (SSA)). D'abord, nous montrons que SSA et MSA sont deux problèmes NP-difficiles. Ensuite, nous fournissons une borne supérieure sur la valeur de la solution optimale de MSA en résolvant SSA dans chaque *slot* et nous montrons que cette borne supérieure est étroite. Ces deux contributions sont importantes en théorie et en pratique. Premièrement, fournir une borne supérieure pour MSA est une étape clé dans la conception des algorithmes d'approximation (Vazirani, 2010). Deuxièmement, résoudre MSA *slot* par *slot* est plus pratique dans les réseaux DCNs car les utilisateurs arrivent *online* comme un flux de données. Donc, nous pouvons utiliser SSA à chaque *slot* de manière indépendante et fournir une solution proche de la solution optimale de MSA. Finalement, nous proposons un algorithme distribué

simple en utilisant la théorie des jeux et l'apprentissage automatique.

2.2.1 Contributions

Les contributions réalisées dans ce chapitre sont les suivantes.

- Nous formulons MSA comme un programme linéaire en nombres entiers.
- Nous étudions la NP-difficulté de SSA et de MSA.
- Nous proposons un algorithme centralisé glouton qui résout SSA pendant chaque *slot* et fournit une borne supérieure étroite sur la valeur de la solution de MSA.
- Nous modélisons MSA comme un jeu non coopératif.
- Nous étudions les équilibres de ce jeu et nous étudions son prix d'anarchie et son prix de stabilité.
- Nous proposons un algorithme totalement distribué inspiré de l'algorithme d'apprentissage bien connu *Win-Stay, Lose-Shift* (Nowak et Sigmund, 1993).

2.2.2 Organisation

Le reste de ce chapitre est organisé comme suit. La section 2.3 présente et analyse une revue de littérature. La section 2.4 présente le modèle de système. La section 2.5 formule le problème. La section 2.6 étudie sa NP-difficulté. La section 2.7 présente les solutions centralisées alors que la section 2.8 présente les solutions distribuées. Les résultats de simulations sont donnés dans la section 2.9. La section 2.10 conclut le chapitre.

2.3 Revue de littérature

Les principaux travaux de recherche liés à MSA sont discutés ci-après. Dans (Mlika *et al.*, 2017e), nous avons proposé des algorithmes heuristiques pour résoudre SSA ; un

travail réalisé pendant la maîtrise en informatique du candidat Z. Mlika. Nous avons montré que ce problème est NP-difficile en utilisant une réduction complexe à partir du problème de satisfaisabilité. Dans ce chapitre, en plus d'étudier SSA, nous étudions sa version *multi-slot*; ce qui n'a pas été fait dans (Mlika *et al.*, 2017e). En outre, nous proposons une démonstration de NP-difficulté qui est plus simple par rapport à celle présentée dans (Mlika *et al.*, 2017e) en utilisant le problème d'activation de lien, ou en anglais *Link Scheduling (LS)* (Halldórsson et Mitra, 2011). Notre nouvelle démonstration (i) peut être utilisée pour démontrer que d'autres problèmes liés à MSA et à LS sont NP-difficiles et (ii) permet d'étudier la difficulté d'approximation (Crescenzi, 1997) de MSA. Dans (Zhou *et al.*, 2015), les auteurs proposent des algorithmes gloutons et d'approximation afin de résoudre le problème d'ordonnancement et d'association des utilisateurs dans les réseaux DCNs avec des stations de base femto-cellules. Leurs approches sont basées sur des algorithmes classiques d'équilibrage de charge min-max (*min-max load balancing* en anglais). Dans (Ao et Psounis, 2017), des algorithmes d'approximation et des algorithmes *online* compétitifs sont proposés pour résoudre le problème d'association des utilisateurs dans les réseaux DCNs. Les algorithmes proposés maximisent le débit total des utilisateurs associés. Dans les deux travaux (Zhou *et al.*, 2015; Ao et Psounis, 2017), en plus d'utiliser des fonctions objectif différentes par rapport à celle utilisée dans ce chapitre, les auteurs utilisent le pire cas d'interférence en supposant qu'une station de base interfère avec toutes les autres, même si elle n'associe aucun utilisateur. En outre, il n'y avait pas une démonstration de NP-difficulté dans (Ao et Psounis, 2017). Dans (Zhou *et al.*, 2015), même si les auteurs ont montré que leur problème est NP-difficile, cette démonstration ne peut pas être utile dans notre cas puisque (i) elle utilise le fait que leur problème est un programme linéaire en nombres entiers et (ii) la fonction objectif de notre problème est différente. Le travail de (Halldórsson et Mitra, 2011) étudie LS sous les contraintes de SINR. Dans ce problème, il existe un certain nombre de liens (un lien est une paire émetteur-récepteur) où chaque émetteur se voit allouer une certaine puis-

sance de transmission. L'objectif de LS est de maximiser le nombre de liens à activer dans un *slot* sous les contraintes de SINR, où un lien est actif lorsqu'on lui alloue une puissance positive. Les auteurs ont étudié LS dans deux cas ; avec puissance de transmission fixe et variable. Ils ont proposé des algorithmes d'approximation efficaces dans chaque cas. Dans (Goussevskaja *et al.*, 2009; Goussevskaja *et al.*, 2014), les auteurs ont étudié LS dans deux scénarios différents où chaque scénario optimise une fonction objectif ; un maximise le nombre de liens à activer dans un *slot* et un minimise le nombre de *slots* à utiliser afin d'activer tous les liens. Le problème MSA étudié dans cette partie de thèse est fondamentalement différent de LS étudié dans (Halldórsson et Mitra, 2011; Goussevskaja *et al.*, 2009; Goussevskaja *et al.*, 2014) puisque dans MSA les paires émetteur-récepteur ne sont pas connues auparavant, c.-à-d., nous cherchons plutôt d'associer les émetteurs (p. ex. les utilisateurs) aux récepteurs (p. ex. les stations de base).

Les travaux de recherche qui étudient des problèmes similaires à MSA d'une manière distribuée sont discutés ci-après. La référence (Shen et Yu, 2014) résout le problème d'association des utilisateurs en se basant sur une méthode de tarification et sur une analyse lagrangienne duale. Les auteurs étudient également ce problème dans le cas des communications à antennes multiples et optimisent les variables de formation de faisceaux. Ils se concentrent sur un modèle de système différent de celui abordé dans le présent chapitre et, de plus, les solutions proposées ne sont pas complètement distribuées. Dans (Wang *et al.*, 2016), les auteurs étudient le problème d'association des utilisateurs et d'allocation des fréquences dans les réseaux DCNs avec des communications à antennes multiples à grande échelle. Une méthode de décomposition est utilisée pour résoudre ce problème. Ensuite, sur la base d'une analyse lagrangienne duale, les auteurs proposent un algorithme distribué. Malgré la nature distribuée de leur algorithme, la quantité d'informations échangées est grande et est proportionnelle au nombre de stations de base et d'utilisateurs. Notre algorithme distribué proposé dans ce chapitre, en contre partie, est totalement distribué avec zéro quantité d'informations échangées. Les

articles (Sohn et Lee, 2016; Ali *et al.*, 2016) étudient le problème d'association des utilisateurs afin d'équilibrer la charge des réseaux DCNs. Ils proposent respectivement des algorithmes de passage de messages et d'apprentissage. La théorie des graphes est utilisée dans (Sohn et Lee, 2016) afin de concevoir l'algorithme de passage de messages. Dans cet algorithme les stations de base et les utilisateurs doivent échanger plusieurs messages avant la convergence. Dans (Ali *et al.*, 2016), les auteurs modélisent le problème en utilisant la théorie de jeu potentiel et proposent des algorithmes d'apprentissage pour atteindre l'équilibre de Nash. Tous les travaux cités ci-dessus (Shen et Yu, 2014; Wang *et al.*, 2016; Sohn et Lee, 2016; Ali *et al.*, 2016) proposent des algorithmes qui ne sont pas totalement distribués ou qui fonctionnent sous différentes hypothèses de modèle de système. Les travaux (Andrews et Dinitz, 2009; Dinitz, 2010; Ásgeirsson et Mitra, 2011) proposent des algorithmes d'apprentissage pour LS. Dans (Andrews et Dinitz, 2009), les auteurs formulent LS comme un jeu non coopératif et étudient sa convergence vers un équilibre de Nash mixte. Le même modèle de système est considéré dans (Dinitz, 2010; Ásgeirsson et Mitra, 2011) où les auteurs ont conçu des algorithmes sans regret qui convergent vers des performances presque optimales. Ils ont utilisé l'algorithme totalement distribué, connu sous le nom *Randomized Weighted Majority (RWM)*.

2.4 Modèle de système

Nous considérons un réseau DCN composé de n stations de base notées par $\mathcal{B} \triangleq \{1, 2, \dots, n\}$. Dans ce réseau, k utilisateurs, notés par l'ensemble $\mathcal{U} \triangleq \{1, 2, \dots, k\}$, transmettent des messages aux stations de base, autrement dit, la transmission sur le lien montant est considérée¹. Les communications entre les utilisateurs et les stations de base se produisent sur un ensemble de trames où chaque trame est composé de m slots.

1. Nous pouvons facilement considérer la transmission sur le lien descendant, après les changements adéquats aux niveaux des formules mathématiques. Néanmoins, la transmission sur le lien descendant est considérée plus tard lorsque nous étudions le scénario distribué.

Nous utilisons $\mathcal{S} \triangleq \{1, 2, \dots, m\}$ pour noter l'ensemble de *slots* disponibles dans une trame quelconque. Chaque utilisateur u dans l'ensemble $\mathcal{U}$ a un paquet à transmettre pendant chaque trame. Nous nous concentrons sur une trame quelconque et nous supposons qu'un de ses *slots* correspond à la durée de transmission d'un paquet². Lorsque l'utilisateur u est associé à une station de base, il transmet son paquet avec une puissance p_u et nécessite un rapport signal-sur-interférence-plus-bruit (SINR) de seuil β_u . L'utilisateur u ne peut correctement transmettre son paquet que si la puissance de son signal est β_u fois supérieure à la puissance de l'interférence plus la puissance du bruit.

En général, l'association des utilisateurs doit se faire sur une longue durée afin d'éviter (i) un transfert intercellulaire inutile ou (ii) l'effet de ping-pong (Xu *et al.*, 2017). Ainsi, le canal sans fil entre les utilisateurs et les stations de base prend en considération l'évanouissement à grande échelle seulement. Cette hypothèse est largement utilisée dans la littérature, p. ex. (Zhuang *et al.*, 2016). Par conséquent, le canal sans fil entre les utilisateurs et les stations de base est supposé fixe pendant toute la durée de la trame en considération, cependant, il peut changer d'une trame à une autre.

Ce chapitre étudie le problème d'association des utilisateurs aux stations de base sur l'ensemble de m *slots*. Ce problème est appelé *multi-slot association* (MSA). Son objectif est de maximiser le nombre d'utilisateurs associés pendant les m *slots* de telle sorte que les exigences en SINR de chaque utilisateur associé soient respectées. Cet objectif maximise la capacité du réseau (Goussevskaia *et al.*, 2009). Évidemment, cette maximisation pénalise certains utilisateurs dans le sens qu'il y aura des utilisateurs qui ne seront pas associés. Nous pouvons surmonter cette difficulté en maximisant une fonction pondérée du nombre d'utilisateurs associés, c.-à-d. à chaque utilisateur est assigné un nombre positif et nous maximisons la somme de ces nombres. Avec un choix précis des poids,

2. Une fois qu'un utilisateur est associé, il est ordonnancé pendant le *slot* correspondant. Autrement dit, il n'y a pas de problème d'ordonnancement. Ce problème est étudié dans les prochains chapitres.

nous pouvons en outre augmenter l'équité des utilisateurs puisque tout le monde sera associé à peu près le même nombre de fois.

2.5 Formulation du problème

Afin de formuler MSA comme un programme linéaire en nombres entiers, nous utilisons les variables binaires suivantes :

$$x_{u \rightarrow b}^s \triangleq \begin{cases} 1, & \text{si } u \text{ est associé à } b \text{ pendant } s, \\ 0, & \text{sinon} \end{cases} \quad (2.1)$$

où $(u, b, s) \in \mathcal{U} \times \mathcal{B} \times \mathcal{S}$.

Nous utilisons la notation matricielle $\mathbf{X}$ pour noter toutes les variables binaires $x_{u \rightarrow b}^s$ pour tout $u \in \mathcal{U}$, $b \in \mathcal{B}$ et $s \in \mathcal{S}$. Le rapport SINR entre l'utilisateur u et la station de base b pendant le *slot* s peut être écrit comme suit :

$$\text{SINR}_{u \rightarrow b}^s(\mathbf{X}) \triangleq \frac{\gamma_u g_{u \rightarrow b} x_{u \rightarrow b}^s}{1 + \sum_{u' \in \mathcal{U} \setminus \{u\}} \sum_{b' \in \mathcal{B} \setminus \{b\}} \gamma_{u'} g_{u' \rightarrow b} x_{u' \rightarrow b'}^s}, \quad (2.2)$$

où $g_{u \rightarrow b}$ est le gain du canal sans fil entre u et b et $\gamma_u \triangleq p_u / \sigma^2$ avec σ^2 est la puissance d'un bruit additif blanc gaussien (*additive white gaussian noise (AWGN)* en anglais).

Maintenant, MSA peut être formulé comme suit :

$$\text{maximiser} \quad \sum_{u \in \mathcal{U}} \sum_{b \in \mathcal{B}} \sum_{s \in \mathcal{S}} x_{u \rightarrow b}^s \quad (2.3a)$$

$$\text{sous les contraintes} \quad \sum_{u \in \mathcal{U}} x_{u \rightarrow b}^s \leq 1, \quad \forall b \in \mathcal{B}, s \in \mathcal{S} \quad (2.3b)$$

$$\sum_{b \in \mathcal{B}} \sum_{s \in \mathcal{S}} x_{u \rightarrow b}^s \leq 1, \quad \forall u \in \mathcal{U} \quad (2.3c)$$

$$\text{SINR}_{u \rightarrow b}^s(\mathbf{X}) \geq \beta_u x_{u \rightarrow b}^s, \quad \forall u \in \mathcal{U}, b \in \mathcal{B}, s \in \mathcal{S} \quad (2.3d)$$

$$x_{u \rightarrow b}^s \in \{0, 1\}, \quad \forall u \in \mathcal{U}, b \in \mathcal{B}, s \in \mathcal{S}. \quad (2.3e)$$

La fonction objectif (2.3a) spécifie le nombre d'utilisateurs associés aux stations de base pendant tous les *slots*. Les contraintes (2.3b) garantissent que, pour un *slot* donné $s \in \mathcal{S}$ et une station de base donnée $b \in \mathcal{B}$, il ne peut y avoir plus d'un utilisateur associé. Les contraintes (2.3c) garantissent que pour un utilisateur donné $u \in \mathcal{U}$, il doit exister au plus une station de base et un *slot* auxquels l'utilisateur u est associé. Les contraintes (2.3d) garantissent que les rapports SINR soient respectés pour les utilisateurs associés. Enfin, les contraintes (2.3e) garantissent que les variables $x_{u \rightarrow b}^s$ sont binaires pour tout $u \in \mathcal{U}, b \in \mathcal{B}, s \in \mathcal{S}$.

Le problème (2.3) est un programme mathématique non-linéaire en raison des contraintes (2.3d). En effet, ces contraintes s'écrivent explicitement comme :

$$\frac{\gamma_u g_{u \rightarrow b} x_{u \rightarrow b}^s}{1 + \sum_{u' \in \mathcal{U} \setminus \{u\}} \sum_{b' \in \mathcal{B} \setminus \{b\}} \gamma_{u'} g_{u' \rightarrow b} x_{u' \rightarrow b'}^s} \geq \beta_u x_{u \rightarrow b}^s. \quad (2.4)$$

Nous pouvons remarquer que le dénominateur ainsi que le côté droit de (2.4) contiennent les variables $x_{u \rightarrow b}^s$. Ainsi, (2.4) contient des termes quadratiques; ce qui la rend non-linéaire. Toutefois, nous pouvons linéariser cette équation en utilisant la technique bigM (Schrijver, 1986). Cette technique consiste à introduire un grand nombre positif M qui va permettre d'enlever les variables $x_{u \rightarrow b}^s$ du côté droit de (2.4). Soit M un nombre positif. Les contraintes (2.4) peuvent s'écrire d'une manière équivalente comme :

$$\frac{\gamma_u g_{u \rightarrow b} x_{u \rightarrow b}^s + M(1 - x_{u \rightarrow b}^s)}{1 + \sum_{u' \in \mathcal{U} \setminus \{u\}} \sum_{b' \in \mathcal{B} \setminus \{b\}} \gamma_{u'} g_{u' \rightarrow b} x_{u' \rightarrow b'}^s} \geq \beta_u. \quad (2.5)$$

En choisissant M soigneusement, nous pouvons voir que (2.4) et (2.5) sont équivalentes. En effet, si $x_{u \rightarrow b}^s = 1$, alors (2.4) et (2.5) sont équivalentes. D'autre part, si $x_{u \rightarrow b}^s = 0$,

alors (2.5) devient

$$\frac{M}{1 + \sum_{u' \in \mathcal{U} \setminus \{u\}} \sum_{b' \in \mathcal{B} \setminus \{b\}} \gamma_{u'} g_{u' \rightarrow b} x_{u' \rightarrow b'}^s} \geq \beta_u. \quad (2.6)$$

Donc, si $M \geq \max_{u,b} \beta_u (1 + \sum_{u' \in \mathcal{U} \setminus \{u\}} \sum_{b' \in \mathcal{B} \setminus \{b\}} \gamma_{u'} g_{u' \rightarrow b})$, alors (2.4) et (2.5) sont équivalentes. Par conséquent, en remplaçant (2.4) par (2.5), le problème (2.3) peut être reformulé comme un programme linéaire en nombres entiers. Théoriquement, cette technique de linéarisation produit deux problèmes équivalents, cependant, sa mise en pratique peut donner lieu à des difficultés numériques (IBM, 2010). Pour assurer la robustesse numérique, nous exprimons les contraintes (2.5) en utilisant des contraintes indicatrices (en anglais *Indicator Constraints* (ICs)) modélisées dans les solveurs modernes comme CPLEX (IBM, 2010). Cette technique de linéarisation est utilisée dans plusieurs parties de cette thèse.

Nous notons que lorsque $m = 1$, autrement dit, un seul *slot* est considéré, le problème (2.3) nous donne un programme linéaire en nombres entiers pour la version *uni-slot* de MSA, appelée *single-slot association* (SSA).

2.6 NP-difficulté

Dans cette section, nous considérons les deux problèmes de décisions correspondant aux SSA et MSA. Nous les notons respectivement par D-SSA et D-MSA et nous montrons qu'il s'agit de deux problèmes NP-complets. Nous commençons par réduire, en temps polynomial, le problème NP-difficile d'activation de liens (abrégé LS) (Halldórson et Mitra, 2011) à D-SSA. La NP-complétude de D-MSA s'ensuit par restriction (Garey et Johnson, 1979), puisque SSA est un cas particulier de MSA. Une démonstration de NP-complétude de D-SSA a été présentée dans (Mlika *et al.*, 2017e) où elle se base sur une réduction polynomiale complexe en utilisant le problème de satisfiabilité. La démon-

tration présentée dans cette thèse est beaucoup plus simple et utilise un problème très connu dans les réseaux sans fil, soit celui d'activation de liens. Notre démonstration, en raison de sa simplicité, peut facilement être utilisée pour montrer que d'autres problèmes connexes sont NP-difficile et aussi pour étudier la difficulté d'approximation (Crescenzi, 1997) de certains problèmes dans les réseaux sans fil.

Dans LS, il existe un seul *slot* et un ensemble de liens où chaque lien représente une paire émetteur-récepteur. La question qui se pose dans LS est : comment trouver le plus grand nombre de liens à activer (ou à sélectionner) tel que le SINR de chaque lien activé soit respecté. Plus formellement, la version de décision de LS (D-LS) est définie comme suit :

Définition 2.6.1 (D-LS) :

- INSTANCE : une matrice $[g_{i \rightarrow j}]$ de taille $\ell \times \ell$ qui représente les gains des canaux sans fil des différents liens, un vecteur $[p_1, \dots, p_\ell]$ qui représente les puissances de transmission des liens, un entier positif θ et un nombre réel positif ε .
- QUESTION : est-ce qu'il existe un sous-ensemble $\mathcal{X}$ de liens, où $\mathcal{X} \subseteq \{1, 2, \dots, \ell\}$, tel que $|\mathcal{X}| \geq \theta$ et

$$\frac{p_i g_{i \rightarrow i}}{1 + \sum_{j \in \mathcal{X} \setminus \{i\}} p_j g_{i \rightarrow j}} \geq \varepsilon, \forall i \in \mathcal{X}. \quad (2.7)$$

D'autre part, la version de décision de SSA (D-SSA) est définie comme suit :

Définition 2.6.2 (D-SSA) :

- INSTANCE : une matrice $[h_{j \rightarrow i}]$ de taille $k \times n$ qui représente les gains des canaux sans fil entre chaque paire utilisateur-station de base (j, i) , un vecteur $[q_1, \dots, q_k]$ qui représente les puissances de transmission des utilisateurs, un entier positif r et un nombre réel positif α .

— QUESTION : est-ce qu'il existe un sous-ensemble $\mathcal{Y}$ de $\{(j, i) : j \in \{1, \dots, k\} \text{ et } i \in \{1, \dots, n\}\}$ tel que $|\mathcal{Y}| \geq r$ et :

$$\frac{q_j h_{j \rightarrow i}}{1 + \sum_{j' \in \mathcal{Y}_j} q_{j'} h_{j' \rightarrow i}} \geq \alpha, \forall (j, i) \in \mathcal{Y}, \quad (2.8)$$

avec $\mathcal{Y}_j \triangleq \{j' : (j', i') \in \mathcal{Y} \text{ et } j' \neq j\}$.

Théorème 2.6.3 *D-SSA est NP-complet.*

Démonstration : étant donné une instance de D-LS, nous construisons une instance de D-SSA comme suit. Soit $n = k \triangleq \ell$, $q_i \triangleq p_i$ pour tout i , $r \triangleq \theta$ et $\alpha \triangleq \max_{i' \neq j'} \{g_{i' \rightarrow j'}\}$. La matrice $[h_{j \rightarrow i}]$ est donnée par :

$$h_{j \rightarrow i} = \begin{cases} \alpha g_{i \rightarrow j} / \varepsilon, & \text{si } i = j \\ g_{i \rightarrow j}, & \text{si } i \neq j \end{cases} \quad (2.9)$$

Clairement, la création de l'instance de D-SSA est faite en temps polynomial. Il reste à montrer que l'instance de D-LS est résolue si et seulement si l'instance de D-SSA est résolue. En d'autres termes, nous montrons qu'il existe un sous-ensemble $\mathcal{X}$ de liens de cardinalité $|\mathcal{X}| \geq \theta$ tel que (2.7) est satisfaite, si et seulement si, il existe un sous-ensemble $\mathcal{Y}$ de $\{(j, i) : i \in \{1, \dots, k\} \text{ et } j \in \{1, \dots, n\}\}$ de taille $|\mathcal{Y}| \geq r$ tel que (2.8) est satisfaite.

D'abord, nous notons que si $i \neq j$, alors $h_{j \rightarrow i} = g_{i \rightarrow j} < \alpha = \max_{i' \neq j'} \{g_{i' \rightarrow j'}\}$. Par conséquent, si $i \neq j$, alors la station de base i ne peut pas être associée à l'utilisateur j . Supposons que D-LS est résolu. En se basant sur (2.7), nous avons :

$$\begin{aligned} & \frac{p_i g_{i \rightarrow i}}{1 + \sum_{j' \in \mathcal{X} \setminus \{i\}} p_{j'} g_{i \rightarrow j'}} \geq \varepsilon, \forall i \in \mathcal{X}, \\ \iff & \frac{q_i \varepsilon h_{j \rightarrow i} / \alpha}{1 + \sum_{j' \in \mathcal{X} \setminus \{i\}} q_{j'} h_{j' \rightarrow i}} \geq \varepsilon, \forall (j, i) \in \mathcal{Y} \text{ et } i = j, \end{aligned}$$

$$\iff \frac{q_i h_{j \rightarrow i}}{1 + \sum_{j' \in \mathcal{X} \setminus \{i\}} q_{j'} h_{j' \rightarrow i}} \geq \alpha, \forall (j, i) \in \mathcal{Y} \text{ et } i = j.$$

Ainsi, $\mathcal{Y} = \mathcal{X} \times \mathcal{X}$ satisfait (2.8). Par conséquent, D-LS est résolu si et seulement si D-SSA est résolu. Comme l'instance de D-SSA est créée en temps polynomial et que LS est NP-difficile, alors D-SSA est NP-difficile. D-SSA appartient aussi à la classe NP puisque, étant donnée une solution de D-SSA, nous pouvons vérifier en temps polynomial qu'elle s'agit bien d'une solution valide. Donc, D-SSA est NP-complet. Ceci prouve le théorème. ■

Comme D-SSA est un cas spécial de D-MSA, nous savons, par restriction (Garey et Johnson, 1979), que MSA est également NP-difficile. Il est aussi NP-complet puisque nous pouvons vérifier une solution de D-MSA en temps polynomial.

2.7 Algorithmes centralisés

Dans cette section, nous proposons un algorithme glouton qui résout MSA de manière centralisée. Cet algorithme se base sur la résolution de SSA pendant chaque *slot*. Nous notons par OPT-MSA (resp. par OPT-SSA(s)) un algorithme optimal qui résout MSA (resp. SSA pendant le *slot* s). Nous proposons de résoudre MSA en utilisant OPT-SSA(s) à chaque *slot* $s = 1, 2, \dots, m$. Nous appelons cet algorithme ALG-MSA et il est décrit comme suit. Pour le premier *slot* $s = 1$, OPT-SSA(1) est exécuté avec tous les utilisateurs et toutes les stations de base. Avant de passer au deuxième *slot* $s = 2$, ALG-MSA met à jour les ensembles des utilisateurs et des stations de base en enlevant ceux déjà associés dans le premier *slot*. ALG-MSA continue son exécution de la même façon pendant les *slots* consécutifs $s = 3, 4, \dots, m$. Soit o (resp. o_s) la valeur optimale retournée par OPT-MSA (resp. par OPT-SSA(s)). Soit $A \triangleq \sum_{s=1}^m o_s$ la valeur de la solution retournée par ALG-MSA. Nous démontrons qu' A est proche de o . Avant de présenter notre analyse, nous discutons les observations suivantes.

Observation 2.7.1 Dans cette observation, nous donnons un exemple pour lequel $o \neq A$. Soit $m = 3$, $k = 6$ et $n = 6$. Une solution pour SSA pourrait être $o_1 = 3$, $o_2 = 1$ et $o_3 = 1$ en associant les paires utilisateur-station de base $(1, 1)$, $(2, 2)$, $(3, 3)$ dans le premier slot, $(4, 4)$ dans le deuxième slot et $(5, 5)$ dans le troisième slot. Par conséquent, $A = 5$. Cependant, une solution pour MSA pourrait être $o = 6$ en associant les paires $(1, 1)$, $(4, 4)$ dans le premier slot, $(2, 2)$, $(5, 5)$ dans le deuxième slot et $(3, 3)$, $(6, 6)$ dans le troisième slot.

Observation 2.7.2 Dans cette observation, nous donnons une condition suffisante pour que $o = A$. Supposons qu'il existe un slot s' tel que $o_s = 0$ pour tout $s \geq s'$. Donc, il existe deux cas : (i) les k utilisateurs sont associés ou (ii) $k - \sum_{s < s'} o_s$ ne peuvent pas être associés pendant aucun slot. Dans les deux cas, o ne peut être supérieur à $\sum_{s < s'} o_s$, ce qui implique, par l'optimalité de o , que $o = \sum_{s < s'} o_s = A$.

Une question naturelle qui se pose maintenant est la suivante : quel est le ratio entre o et A ? En répondant à cette question, nous dérivons une borne supérieure sur o en utilisant le principe des tiroirs. D'abord, en se basant sur l'observation 2.7.2, nous pouvons supposer, sans perte de généralité, que $o_s \neq 0$ pour tout $s = 1, 2, \dots, m$.

Lemme 2.7.3 Pour tout $s = 1, 2, \dots, m$, nous avons :

$$o_s \geq \frac{o - \sum_{s'=1}^{s-1} o_{s'}}{m}.$$

Démonstration : à n'importe quel slot s , nous pouvons trouver au moins $o - \sum_{s'=1}^{s-1} o_{s'}$ utilisateurs qui ne sont pas associés pendant les $s - 1$ premiers slots par ALG-MSA mais ils sont associés par OPT-MSA pendant les m slots. Par conséquent, selon le principe des tiroirs, un des m slots de la solution retournée par OPT-MSA doit contenir au moins $(o - \sum_{s'=1}^{s-1} o_{s'})/m$ utilisateurs car, sinon, nous ne pourrions pas obtenir o pendant les m slots. Puisque, par définition, OPT-SSA(s) associe le nombre maximum d'utilisateurs restants pendant le slot s , donc $o_s \geq (o - \sum_{s'=1}^{s-1} o_{s'})/m$. Ceci prouve le lemme. ■

Lemme 2.7.4 Pour tout $s = 1, 2, \dots, m$, nous avons :

$$\sum_{s'=1}^s o_{s'} \geq o \left(1 - \left(1 - \frac{1}{m}\right)^s\right).$$

Démonstration : nous démontrons le lemme en utilisant un raisonnement par récurrence sur s . Pour $s = 1$, nous savons par le lemme 2.7.3 que $o_1 \geq o/m$ et donc la base de la récurrence est vraie. Pour démontrer l'hypothèse de récurrence, supposons que le résultat est vrai pour s et montrons-le pour $s + 1$. L'hypothèse de récurrence dit que $\sum_{s'=1}^s o_{s'} \geq o(1 - (1 - \frac{1}{m})^s)$. Nous avons :

$$\begin{aligned} o_{s+1} &\stackrel{(a)}{\geq} \frac{1}{m} \left(o - \sum_{s'=1}^s o_{s'} \right) \\ \Rightarrow \sum_{s'=1}^{s+1} o_{s'} &\geq \frac{1}{m} \left(o - \sum_{s'=1}^s o_{s'} \right) + \sum_{s'=1}^s o_{s'} \\ \Rightarrow \sum_{s'=1}^{s+1} o_{s'} &\geq \frac{1}{m} o + \left(1 - \frac{1}{m}\right) \sum_{s'=1}^s o_{s'} \\ \Rightarrow \sum_{s'=1}^{s+1} o_{s'} &\stackrel{(b)}{\geq} \frac{1}{m} o + o \left(1 - \frac{1}{m}\right) \left(1 - \left(1 - \frac{1}{m}\right)^s\right) \\ \Rightarrow \sum_{s'=1}^{s+1} o_{s'} &\geq o \left(1 - \left(1 - \frac{1}{m}\right)^{s+1}\right), \end{aligned} \tag{2.10}$$

où (a) est due au lemme 2.7.3 et (b) est due à l'hypothèse de récurrence. En utilisant la base de récurrence ainsi que (2.10), nous concluons que $\sum_{s'=1}^s o_{s'} \geq o(1 - (1 - 1/m)^s)$ pour tout s . Ceci complète la démonstration. ■

Théorème 2.7.5 Nous avons

$$A \leq o \leq \frac{A}{1 - (1 - 1/m)^m} \leq \frac{A}{1 - 1/e},$$

où e représente le nombre d'Euler.

Démonstration : puisque $A = \sum_{s=1}^m o_s$, la démonstration s'ensuit à partir du lemme 2.7.4

pour $s = m$ et à partir du fait suivant : pour tout nombre réel x , nous avons $1 - x \leq e^{-x}$. En choisissant $x = 1/m$, nous obtenons

$$A \leq O \leq \frac{A}{1 - (1 - 1/m)^m} \leq \frac{A}{1 - 1/e}.$$

Ceci complète la démonstration. ■

Le théorème 2.7.5 établit une borne supérieure sur O . Il montre que, dans le pire cas, le ratio O/A est majoré par 1,582 (puisque $1/(1 - 1/e)$ est presque égal à 1,582). Cette borne supérieure est très intéressante parce qu'elle est indépendante de k , n et m . Bien que SSA et MSA soient deux problèmes NP-difficiles, et donc calculer leurs solutions optimales est NP-difficile (Vazirani, 2010), cette borne supérieure peut être utilisée pour concevoir des algorithmes d'approximation efficaces pour MSA.

Observation 2.7.6 Dans cette observation, nous donnons un exemple d'instance du problème MSA afin de montrer l'étroitesse de la borne supérieure. La borne supérieure donnée dans le théorème 2.7.5 est égale à $x_m \triangleq 1/(1 - (1 - 1/m)^m)$, qui tend vers $1/(1 - 1/e)$ lorsque m tend vers l'infini. Pour prouver que x_m est une borne supérieure étroite, il suffit de trouver une instance de MSA pour laquelle $O = x_m A$. Soit $k = 4$, $n = 2$, $m = 2$, $\beta_u = 1$, $\gamma_u = 1$ pour tout u et la matrice des gains des canaux sans fil donnée par :

$$G = \begin{pmatrix} 5 & 2 \\ 3 & 5 \\ 2 & 3 \\ 2 & 6 \end{pmatrix}.$$

D'abord, $x_2 = 4/3$. La solution optimale pour cette instance est $O = 4$ et elle peut être obtenue en associant l'utilisateur 1 à la station de base 1 et l'utilisateur 3 à la station de base 2 pendant le premier slot et l'utilisateur 2 à la station de base 1 et l'utilisateur 4 à la station

de base 2 pendant le deuxième slot. Cependant, ALG-MSA peut associer l'utilisateur 1 à la station de base 1 et l'utilisateur 2 à la station de base 2 pendant le premier slot et l'utilisateur 3 seul (ou l'utilisateur 4 seul) à n'importe quelle station de base dans le deuxième slot. Ainsi, $A = 3$ et donc $O = (4/3)A = x_2A$. Cela prouve l'étroitesse de la borne supérieure obtenue par ALG-MSA.

2.8 Algorithmes distribués

L'objectif de cette partie est de résoudre MSA en développant un algorithme distribué afin de minimiser la quantité d'informations échangées et de réduire la dépendance à un point central qui constitue souvent un point de défaillance unique. Une partie de ce travail a été faite pendant la maîtrise en informatique du candidat Z. Mlika. Dans cette thèse de doctorat, nous analysons davantage les solutions proposées et nous les améliorons. Pour ce faire, nous modifions le modèle de système adopté dans la section précédente en considérant la transmission sur le lien descendant. Cette modification permet d'implémenter l'algorithme proposé au niveau des stations de base d'une manière distribuée et de réduire la quantité d'informations échangées. En effet, la considération de la transmission sur le lien montant oblige les utilisateurs d'échanger des informations afin de savoir à quelles stations de base ils sont associés. Par conséquent, nous modifions légèrement le modèle de système présenté précédemment. Cette modification est appliquée au niveau de (2.2) comme suit :

$$\text{SINR}_{b \rightarrow u}^s(\mathbf{X}) \triangleq \frac{\gamma_b g_{b \rightarrow u} x_{b \rightarrow u}^s}{1 + \sum_{b' \in \mathcal{B} \setminus \{b\}} \sum_{u' \in \mathcal{U} \setminus \{u\}} \gamma_{b'} g_{b' \rightarrow u} x_{b' \rightarrow u'}^s}, \quad (2.11)$$

où $g_{b \rightarrow u}$ est le gain du canal sans fil entre l'utilisateur u et la station de base b et $\gamma_b \triangleq p_b / \sigma^2$ et p_b est la puissance de la station de base b . Les autres paramètres sont définis de manière similaire en inversant u et b .

2.8.1 Un jeu non-coopératif

Afin de résoudre MSA de manière distribuée, nous le modélisons en utilisant la théorie des jeux. Nous définissons la matrice d'association $\mathbf{A} = [a_b^s]$ pour tout b et s , où $a_b^s \in \mathcal{U}$, c'est-à-dire, $a_b^s = u$ si et seulement si u est associé à b pendant s . En utilisant cette matrice d'association, le SINR entre b et u pendant s peut se réécrire comme suit :

$$\text{SINR}_{b \rightarrow u}^s(\mathbf{A}) \triangleq \frac{\gamma_b g_{b \rightarrow u} \mathbb{1}_{\{a_b^s = u\}}}{1 + \sum_{u' \neq u} \sum_{b' \in \mathcal{B}} \gamma_{b'} g_{b' \rightarrow u} \mathbb{1}_{\{a_{b'}^s = u'\}}}, \quad (2.12)$$

où $\mathbb{1}_{\{a_b^s = u\}}$ est la fonction caractéristique qui est égale à 1 si et seulement si u est associé à b pendant s . Soit $\mathcal{G}$ le jeu non coopératif suivant :

- l'ensemble des joueurs est $\mathcal{B} \times \mathcal{S}$ qui représente le produit cartésien des stations de base et des *slots*, c'est-à-dire, un joueur est la paire (b, s) ;
- l'ensemble des actions $\mathcal{A}_{(b,s)}$ disponibles pour le joueur (b, s) est donné par l'ensemble des utilisateurs plus une action de silence (ou de non-transmission), c.-à-d. $\mathcal{A}_{(b,s)} \triangleq \mathcal{U} \cup \{0\}$ avec 0 l'action de silence. Le profil d'action de tous les joueurs est donné par le vecteur $\mathbf{a} \triangleq [a_{11}, \dots, a_{1m}, \dots, a_{n1}, \dots, a_{nm}]$ qui peut être vu comme la concaténation des lignes de $\mathbf{A}$; et
- la récompense du joueur (b, s) est donnée par la fonction $r_{(b,s)}(\mathbf{a}) : \mathcal{A} \triangleq \mathcal{A}_1 \times \dots \times \mathcal{A}_{nm} \mapsto \{\delta_1, \delta_2, \delta_3\}$:

$$r_{(b,s)}(\mathbf{a}) = r_{(b,s)}(\mathbf{a}_{(b,s)}, \mathbf{a}_{-(b,s)}) \triangleq \begin{cases} \delta_1, & \text{si } C_1 \\ \delta_2, & \text{si } C_2, \\ \delta_3, & \text{si } C_3 \end{cases} \quad (2.13)$$

où $\delta_1 > \delta_2 > \delta_3$ et $\mathbf{a}_{-(b,s)}$ représente le profil d'actions $\mathbf{a}$ en enlevant $\mathbf{a}_{(b,s)}$, c.-à-d., si $\mathbf{a} \triangleq [a_1, a_2, \dots, a_n]$ alors $\mathbf{a}_{-i} \triangleq [a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n]$ et $\mathbf{a} = [a_i, \mathbf{a}_{-i}]$. Les conditions C_1 -

C_3 sont données par :

- C_1 : joueur (b, s) a choisi de s'associer à u et son rapport SINR est satisfait et aucun autre joueur (b', s') n'a choisi u , c.-à-d., $a_{(b,s)} \neq 0$ et $\text{SINR}_{b \rightarrow a_{(b,s)}}^s(\mathbf{A}) \geq \beta_b$ et $\forall (b', s') \neq (b, s), a_{(b,s)} \neq a_{(b',s')}$;
- C_2 : joueur (b, s) a choisi de ne pas s'associer; c.-à-d., $a_{(b,s)} = 0$; et
- C_3 : joueur (b, s) a choisi de s'associer à u et son rapport SINR n'est pas satisfait ou un autre joueur (b', s') a choisi le même utilisateur u , c.-à-d., $a_{(b,s)} \neq 0$ et $\text{SINR}_{b \rightarrow a_{(b,s)}}^s(\mathbf{A}) < \beta_b$ ou $\exists (b', s') \neq (b, s) : a_{(b,s)} = a_{(b',s')}$.

Dans ce qui suit, nous utilisons les définitions suivantes.

Définition 2.8.1 (Équilibre de Nash pur) : Un équilibre de Nash pur est un profil d'actions $\mathbf{a}^* \triangleq (a_{(b,s)}^*, \mathbf{a}_{-(b,s)}^*)$ de telle sorte que pour toute action $a'_{(b,s)}$, nous avons $r_{(b,s)}(a_{(b,s)}^*, \mathbf{a}_{-(b,s)}^*) \geq r_{(b,s)}(a'_{(b,s)}, \mathbf{a}_{-(b,s)}^*)$.

En théorie des jeux, la meilleure réponse est un profil d'actions qui produit le résultat le plus favorable pour un joueur, étant donné les actions des autres joueurs (Fudenberg et Tirole, 1991).

Définition 2.8.2 (Meilleure réponse) : Soit $\mathcal{BR}(a_{-(b,s)})$ l'ensemble des meilleures réponses du joueur (b, s) sachant que les autres joueurs choisissent $\mathbf{a}_{-(b,s)}$. L'action $a_{(b,s)}^*$ appartient à $\mathcal{BR}(a_{-(b,s)})$ si et seulement si, pour tout $a_{(b,s)}$, nous avons $r_{(b,s)}(a_{(b,s)}^*, \mathbf{a}_{-(b,s)}) \geq r_{(b,s)}(a_{(b,s)}, \mathbf{a}_{-(b,s)})$.

Lorsque chaque joueur (b, s) joue une meilleure réponse, nous appelons cet algorithme dynamique de meilleure réponse (BRD).

Théorème 2.8.3 *Le jeu $\mathcal{G}$ n'admet pas d'équilibre de Nash.*

Démonstration : nous prouvons le théorème en construisant un contre-exemple; une instance du jeu qui n'admet pas d'équilibre de Nash. Soit $m = 1, k = n = 3, \gamma_b = 4$ et $\beta_b = 2$

pour tout b . Dans ce cas, comme il n'existe qu'un seul slot, une station de base peut être associée à un seul utilisateur. Soit la matrice des gains des canaux sans fil suivante :

$$\mathbf{G} = \begin{pmatrix} 1 & 3/10 & 1/4 \\ 1/4 & 1 & 3/10 \\ 3/10 & 1/4 & 1 \end{pmatrix}. \quad (2.14)$$

En testant tous les profils d'actions possibles, nous pouvons montrer que cette instance de $\mathcal{G}$ n'admet pas d'équilibre de Nash (en fait les joueurs vont boucler). Ceci prouve le théorème. ■

Bien que le jeu $\mathcal{G}$ n'admette pas d'équilibre de Nash en général, nous observons dans les simulations qu'il admet des équilibres de Nash dans la majorité des réalisations. Cependant, nous montrons dans ce qui suit qu'en modifiant $\mathcal{G}$, nous pouvons garantir l'existence d'au moins un équilibre de Nash. Ceci est montré dans le théorème suivant. Soit $\mathcal{G}_1$ le jeu modifié suivant. Les actions du joueur (b, s) sont $\mathcal{A}_{(b,s)} = \mathcal{U}$ et la récompense est la suivante :

$$r_{(b,s)}(\mathbf{a}_{(b,s)}, \mathbf{a}_{-(b,s)}) \triangleq \begin{cases} \delta_1, & \text{si } \text{SINR}_{b \rightarrow \mathbf{a}_{(b,s)}}^s(\mathbf{A}) \geq \beta_b \\ \delta_2, & \text{si } \text{SINR}_{b \rightarrow \mathbf{a}_{(b,s)}}^s(\mathbf{A}) < \beta_b \\ \delta_3, & \text{si } \exists (b', s') \neq (b, s) : \mathbf{a}_{(b,s)} = \mathbf{a}_{(b',s')} \end{cases}. \quad (2.15)$$

Théorème 2.8.4 *Le jeu $\mathcal{G}_1$ admet au moins un équilibre de Nash.*

Démonstration : la preuve est de montrer que BRD converge vers un équilibre de Nash. Supposons que nous exécutons BRD à partir d'un profil d'actions arbitraire $\mathbf{a}$ tel que $r_{(b,s)}(\mathbf{a}) \in \{\delta_1, \delta_2, \delta_3\}$ pour tout joueur (b, s) . Deux cas peuvent être distingués :

1. $k \geq nm$ (au moins autant d'actions que de joueurs) : dans ce cas, BRD oblige les joueurs qui ont une récompense de δ_3 de changer leurs actions pour obtenir δ_1

ou δ_2 . Les joueurs réussissent à changer leurs actions car il y a plus d'actions que de joueurs. Par conséquent, à la fin de la première itération de BRD, chaque joueur transmet à un utilisateur différent et il n'y a donc pas de collision. Après, les joueurs qui ont un gain de δ_2 appliquent BRD pour trouver un utilisateur inactif (non choisi par les autres joueurs). Si $k = nm$, alors il n'y a pas d'utilisateurs inactifs et aucun des joueurs ne peut dévier. Par conséquent, un équilibre de Nash est atteint. Si $k > nm$, alors chaque joueur peut trouver un utilisateur inactif et y s'associer. Donc, un équilibre de Nash est atteint.

2. $k < nm$ (plus de joueurs que d'actions) : dans ce cas, il y a certains joueurs qui transmettent aux mêmes utilisateurs (c.-à-d., collision). Donc, les joueurs qui reçoivent une récompense de δ_3 ne peuvent pas améliorer leur performance en exécutant BRD. Par conséquent, un équilibre de Nash est déjà atteint.

Cela prouve que BRD conduit à un équilibre où aucun des joueurs n'a l'incitation à dévier. Par conséquent, le jeu $\mathcal{G}_1$ admet au moins un équilibre de Nash. Ceci prouve le théorème. ■

Le jeu $\mathcal{G}_1$ même s'il admet un équilibre, donne des mauvais résultats puisqu'il y aura beaucoup de collisions dans le réseau. Dans ce qui suit, nous nous concentrons sur la résolution du jeu $\mathcal{G}$ en proposant des algorithmes distribués.

2.8.2 Algorithmes proposés

D'abord nous présentons l'algorithme totalement distribué (appelé mWSLS). mWSLS est basé sur une version adaptée de la règle d'apprentissage *Win-Stay, Lose-Shift* (WSLS) (Nowak et Sigmund, 1993). Ensuite, nous présentons trois algorithmes heuristiques distribués, BRD, max-SINR et RWM, qui servent à comparer l'algorithme mWSLS. Nous proposons d'adapter ces trois algorithmes puisqu'ils ont été utilisés pour résoudre des problèmes très similaires à MSA.

2.8.2.1 L'algorithme mWSLS

mWSLS est divisé en trois phases : la première itération, le processus d'apprentissage et la terminaison.

Première itération

Pendant la première phase, chaque joueur (b, s) exécute indépendamment mWSLS. Le joueur (b, s) commence mWSLS en choisissant une action aléatoire $a_{(b,s)} = u$ et s'associe à l'utilisateur correspondant. L'utilisateur u calcule son SINR et renvoie à b un seul bit pour indiquer si son SINR est supérieur au seuil ou non. Par conséquent, le joueur (b, s) calcule sa récompense $r_{(b,s)}$, comme suit :

$$r_{(b,s)}(a_{(b,s)}, \mathbf{a}_{-(b,s)}) = \begin{cases} \delta_1, & \text{si } bit = 1 \\ \delta_2, & \text{si } a_{(b,s)} = 0 \\ \delta_3, & \text{si } bit = 0, \end{cases} \quad (2.16)$$

où *bit* est le bit retourné par l'utilisateur u . Précisément, le joueur (b, s) doit prendre trois décisions : (i) s'il reçoit un bit de retour égal à 1, alors il met à jour sa récompense à δ_1 , (ii) s'il reçoit un bit de retour égal à 0, alors il met à jour sa récompense à δ_3 et (iii) s'il ne reçoit rien, c'est-à-dire qu'il choisit l'action de silence $a_{(b,s)} = 0$, alors il met à jour sa récompense à δ_2 . Sur la base des retours reçus et des récompenses calculées, le joueur qui choisit de transmettre dispose d'une information sur la qualité de l'action choisie alors que le joueur qui choisit le silence n'a pas de nouvelle information. Par conséquent, chaque fois qu'un joueur découvre une nouvelle information, il doit l'exploiter pour jouer une meilleure action dans le futur. Nous associons à chaque joueur (b, s) un vecteur de probabilité $\pi_{(b,s)} = [\pi_{(b,s)}^0, \pi_{(b,s)}^1, \dots, \pi_{(b,s)}^k]$ de taille $k + 1$. Chaque élément $\pi_{(b,s)}^u$ pour tout $u \in \mathcal{U} \cup \{0\}$ correspond à la probabilité de jouer l'action $a_{(b,s)} = u$ par le joueur

(b, s) . Le pseudo-code de mWSLS est donné dans l'algorithme 1.

Algorithme 1 L'algorithme totalement distribué mWSLS pour le joueur (b, s)

```

1: si  $r_{(b,s)}(a_{(b,s)}, a_{-(b,s)}) = \delta_1$  alors
2:    $\pi_{(b,s)}^{a_{(b,s)}} \leftarrow \tau + (1 - \tau)\pi_{(b,s)}^{a_{(b,s)}}$ 
3:   pour  $j$  dans  $\mathcal{A}_{(b,s)}$  faire
4:     si  $j \neq a_{(b,s)}$  alors
5:        $\pi_{(b,s)}^j = (1 - \tau)\pi_{(b,s)}^j$ 
6:     fin si
7:   fin pour
8: sinon si  $r_{(b,s)}(a_{(b,s)}, a_{-(b,s)}) = \delta_3$  alors
9:    $\pi_{(b,s)}^{a_{(b,s)}} \leftarrow \pi_{(b,s)}^{a_{(b,s)}} - \epsilon$ 
10:   $\pi_{(b,s)}^0 = \pi_{(b,s)}^0 + \epsilon$ 
11: fin si
12: retourner  $a_{(b,s)}$ 

```

Processus d'apprentissage

Une fois que les joueurs ont acquis tous les bits retournés par les utilisateurs et les ont transformés en récompenses, ils procèdent à la mise à jour de leurs vecteurs de probabilité, comme indiqué dans l'algorithme 1. Ces mises à jour constituent l'élément central du processus d'apprentissage. Dans l'algorithme WSL S original, chaque joueur commence par jouer une action aléatoire. Si l'action jouée génère une récompense plus élevée, elle est considérée comme une action gagnante et le joueur continue à la jouer à la prochaine itération. Sinon, c'est une action perdante et le joueur doit dévier dans l'espoir d'améliorer sa récompense. WSL S est largement et efficacement appliqué lorsque les récompenses sont booléennes (0 ou 1). Cependant, cet algorithme d'apprentissage doit être adapté lorsque les récompenses sont finies mais pas booléennes comme dans notre cas.

Lorsque la récompense est égale à δ_1 , l'action jouée dans l'itération en cours est considérée comme une action gagnante. Par conséquent, la probabilité de la jouer doit s'améliorer afin d'augmenter les chances de converger vers cette action à la fin du processus

d'apprentissage. Précisément, la probabilité correspondante à l'action gagnante est mise à jour comme suit :

$$\pi_{(b,s)}^u(t+1) = \pi_{(b,s)}^u(t) + \tau(1 - \pi_{(b,s)}^u(t)), \text{ pour tout } u \in \mathcal{U}, \quad (2.17)$$

où t est l'indice de l'itération actuelle³ et τ est appelé le facteur gagnant par lequel la probabilité de choisir l'action gagnante à la prochaine itération est augmentée. Afin d'avoir la somme des probabilités égale à 1, les autres probabilités sont mises à jour comme suit :

$$\pi_{(b,s)}^{u'}(t+1) = \pi_{(b,s)}^{u'}(t) - \tau\pi_{(b,s)}^{u'}(t), \text{ pour tout } u' \neq u \text{ et } u \in \mathcal{U}. \quad (2.18)$$

Lorsque la récompense est égale à δ_2 (c'est-à-dire que l'action de silence est jouée), le joueur n'effectue aucune mise à jour de probabilité car aucun bit n'est reçu.

Lorsque la récompense est égale à δ_3 , l'action choisie dans l'itération en cours est une action perdante. La probabilité de la jouer à la prochaine itération est réduite et la probabilité de rester silencieux est augmentée. Cette stratégie d'apprentissage est motivée par le fait que lorsqu'un joueur joue plusieurs actions perdantes, il est préférable de le forcer à apprendre à jouer l'action de silence. Le vecteur de probabilité est donc mis à jour comme suit :

$$\pi_{(b,s)}^u(t+1) = \pi_{(b,s)}^u(t) - \epsilon, \text{ pour tout } u \in \mathcal{U}, \quad (2.19)$$

$$\pi_{(b,s)}^0(t+1) = \pi_{(b,s)}^0(t) + \epsilon, \quad (2.20)$$

où ϵ est appelé le facteur perdant. mWSLS doit toujours vérifier que $0 \leq \pi_{(b,s)}^u \leq 1$ et $\sum_u \pi_{(b,s)}^u = 1$ pour tout (b, s) et k .

3. L'indice d'une itération est uniquement utilisé lorsque nécessaire.

Terminaison

Chaque joueur termine l'exécution de mWSLS lorsque l'un des critères suivants est rempli : (i) un nombre maximal T_{mWSLS} d'itérations est atteint ou (ii) la probabilité de jouer l'action de silence est soit très proche de un ou très proche de zéro. Un coefficient de tolérance doit être défini pour déterminer si la valeur de probabilité est très proche de un ou non. Dans le cas où une ou plusieurs actions gagnantes sont trouvées, le joueur peut choisir l'action qui donne la probabilité la plus élevée.

2.8.2.2 L'algorithme BRD

L'algorithme BRD peut être appliqué à $\mathcal{G}$. Malheureusement, comme $\mathcal{G}$ n'admet pas toujours un équilibre de Nash, il n'est pas garanti mathématiquement que BRD va converger. BRD est exécuté avec un profil d'actions aléatoire $\mathbf{a}$ pendant T_{BRD} itérations (T_{BRD} est utilisé pour garantir que BRD s'arrête). Le profil d'actions est déterminé à chaque itération en calculant l'ensemble des meilleures réponses de chaque joueur. À chaque itération, le joueur (b, s) met à jour son action en fonction des meilleures réponses précédemment calculées. Ensuite, il choisit la meilleure réponse. Ce processus se répète jusqu'à ce que tous les joueurs mettent à jour leurs actions. Un pseudo-code de BRD est donné dans l'algorithme 2.

2.8.2.3 L'algorithme max-SINR

Une autre façon de résoudre le jeu $\mathcal{G}$ consiste à associer chaque joueur avec l'utilisateur qui lui donne la valeur SINR la plus élevée, appelée max-SINR (3GPP, 2013). max-SINR peut être implémenté de manière distribuée comme suit. Chaque joueur envoie des signaux pilotes à tous les utilisateurs pour estimer les gains des canaux sans fil. Sur la base de ces informations, le joueur reçoit les SINR qui sont renvoyés par les utilisateurs; en supposant que tous les autres joueurs transmettent leurs signaux pilotes.

Algorithme 2 L'algorithme distribué de la meilleure réponse BRD

```

1: pour  $(b, s)$  dans  $\mathcal{B} \times \mathcal{S}$  faire
2:   Joueur  $(b, s)$  choisit une action aléatoire  $a_{(b,s)}$  et calcule sa récompense  $r_{(b,s)}$ 
3: fin pour
4:  $i \leftarrow 0$ 
5: tant que  $i < T_{BRD}$  ou  $a$  n'est pas un équilibre de Nash faire
6:    $\mathcal{BR} \leftarrow \emptyset$ 
7:   pour  $(b, s)$  dans  $\mathcal{B} \times \mathcal{S}$  faire
8:     pour  $u$  dans  $\mathcal{A}_{(b,s)}$  faire
9:       si  $r_{(b,s)}(u, a_{-(b,s)}) > r_{(b,s)}(a_{(b,s)}, a_{-(b,s)})$  alors
10:         $\mathcal{BR} \leftarrow \{u\}$ 
11:         $a_{(b,s)} \leftarrow u$ 
12:       sinon si  $r_{(b,s)}(u, a_{-(b,s)}) = r_{(b,s)}(a_{(b,s)}, a_{-(b,s)})$  alors
13:         $\mathcal{BR} \leftarrow \mathcal{BR} \cup \{u\}$ 
14:       fin si
15:     fin pour
16:      $a_{(b,s)} \leftarrow u$ 
17:   fin pour
18:    $i \leftarrow i + 1$ 
19: fin tant que
20: retourner  $a$ 

```

Ensuite, le joueur s'associe à l'utilisateur avec la valeur SINR la plus élevée et diffuse son choix. L'algorithme continue avec le joueur suivant jusqu'à ce que tous les joueurs aient sélectionné un utilisateur. La dernière étape de l'algorithme exige que tous les joueurs qui ne satisfont pas à leurs exigences SINR restent silencieux. Le pseudocode de max-SINR est donné dans l'algorithme 3.

2.8.2.4 L'algorithme RWM

Nous utilisons l'algorithme *Randomized Weighted Majority* (RWM) pour résoudre le jeu $\mathcal{G}$. Cet algorithme est utilisé dans (Ásgeirsson et Mitra, 2011) pour résoudre LS. Il fonctionne comme suit. On assigne à chaque paire joueur-utilisateur $((b, s), u)$ (ce qui correspond à un lien dans LS) deux poids $w_{b \rightarrow u}^s = [w_{b \rightarrow u}^{s1}, w_{b \rightarrow u}^{s2}]$ où $w_{b \rightarrow u}^{s1}$ et $w_{b \rightarrow u}^{s2}$ corres-

Algorithme 3 L'algorithme distriubé max-SINR

```

1: Tous les joueurs diffusent des signaux pilotes et reçoivent les SINR
2:  $\mathcal{A} \leftarrow \mathcal{U}$ 
3: pour  $(b, s)$  dans  $\mathcal{B} \times \mathcal{S}$  faire
4:    $a_{(b,s)} \leftarrow \arg \max_{u \in \mathcal{A}} \text{SINR}_{b \rightarrow u}^s$ 
5:    $\mathcal{A} \leftarrow \mathcal{A} \setminus \{a_{(b,s)}\}$ 
6:   si  $\text{SINR}_{b \rightarrow a_{(b,s)}}^s < \beta_b$  alors
7:      $a_{(b,s)} \leftarrow 0$ 
8:   fin si
9: fin pour
10: retourner  $a$ 

```

pondent aux poids de silence et de transmission respectivement. RWM est exécuté pendant T_{RWM} itérations. À chaque itération, la paire $((b, s), u)$ choisit le silence ou la transmission avec probabilité $\pi_{b \rightarrow u}^{s1} = w_{b \rightarrow u}^{s1} / (w_{b \rightarrow u}^{s1} + w_{b \rightarrow u}^{s2})$ ou $\pi_{b \rightarrow u}^{s2} = w_{b \rightarrow u}^{s2} / (w_{b \rightarrow u}^{s1} + w_{b \rightarrow u}^{s2})$, respectivement. Les poids ne sont mis à jour que si $((b, s), u)$ choisit de transmettre. Dans ce cas, si le SINR est supérieur au seuil, le poids de silence est multiplié par 1/2, sinon le poids de transmission est multiplié par 1/2. Le pseudo-code de RWM est donné dans l'algorithme 4.

Algorithme 4 L'algorithme totalement distribué RWM pour le joueur (b, s)

```

1: pour  $t \leftarrow 1$  à  $T_{RWM}$  faire
2:   La paire  $((b, s), u)$  joue une action aléatoire selon la probabilité  $\pi_{((b,s),u)} = [\pi_{b \rightarrow u}^{s1}, \pi_{b \rightarrow u}^{s2}]$ 
3:   si  $((b, s), u)$  a choisi de transmettre alors
4:     si  $\text{SINR}_{b \rightarrow u}^s \geq \beta_b$  alors
5:        $w_{b \rightarrow u}^{s1} \leftarrow (1/2)w_{b \rightarrow u}^{s1}$ 
6:     sinon
7:        $w_{b \rightarrow u}^{s2} \leftarrow (1/2)w_{b \rightarrow u}^{s2}$ 
8:     fin si
9:      $\pi_{b \rightarrow u}^{s1} \leftarrow w_{b \rightarrow u}^{s1} / (w_{b \rightarrow u}^{s1} + w_{b \rightarrow u}^{s2})$ 
10:     $\pi_{b \rightarrow u}^{s2} \leftarrow w_{b \rightarrow u}^{s2} / (w_{b \rightarrow u}^{s1} + w_{b \rightarrow u}^{s2})$ 
11:   fin si
12: fin pour
13: retourner L'action de  $((b, s), u)$  selon  $\pi_{((b,s),u)}$ 

```

2.8.2.5 Comparaison des algorithmes distribués

Dans cette partie, nous évaluons la complexité asymptotique et la quantité d'informations échangées pour les algorithmes proposés. Quant à max-SINR, il exige que chaque joueur trouve l'utilisateur ayant la valeur SINR maximale qui donne une complexité de $\mathcal{O}(k)$. Donc, au total, max-SINR a une complexité de $\mathcal{O}(knm)$. En ce qui concerne BRD, la complexité est de $\mathcal{O}(kT_{BRD})$ pour chaque joueur, où T_{BRD} est le nombre maximal d'itérations utilisées pour garantir sa convergence. Lorsque le jeu admet un équilibre de Nash, ce qui se produit très souvent, la complexité de BRD est de $\mathcal{O}(kT_{Nash})$ pour chaque joueur, où T_{Nash} est le temps nécessaire pour trouver un équilibre de Nash. Par conséquent, la complexité totale de BRD est $\min(\mathcal{O}(knmT_{BRD}), \mathcal{O}(knmT_{Nash}))$. Quant à mWSLS, puisqu'il s'agit d'un algorithme totalement distribué, il est implémenté dans chaque joueur séparément. Donc, à chaque itération, chaque joueur requiert $\mathcal{O}(k)$ itérations, y compris une addition et $k+1$ multiplications, comme le montre l'algorithme 1. Globalement, mWSLS a besoin de $\mathcal{O}(kT_{mWSLS})$ itérations pour chaque joueur, où T_{mWSLS} est le nombre maximal d'itérations utilisées par mWSLS. Enfin, l'algorithme RWM nécessite seulement $\mathcal{O}(1)$ itérations pour chaque joueur, ce qui donne une complexité totale de $\mathcal{O}(T_{RWM})$ pour chaque joueur, où T_{RWM} est le nombre maximal d'itérations utilisées par RWM.

La quantité d'informations échangées entre les joueurs de max-SINR, BRD, RWM, et mWSLS est calculée comme suit. Pour max-SINR, les joueurs doivent diffuser leurs actions les unes après les autres, sauf le dernier joueur. Cela nécessite $m(n-1)$ échanges qui est approximativement proportionnel à $\mathcal{O}(nm)$. Pour BRD, la quantité d'informations échangées est proportionnelle au $T_{min} \triangleq \min(T_{BRD}, T_{Nash})$ puisque tous les joueurs doivent diffuser leurs actions à chaque itération, soit T_{BRD} fois, ou jusqu'à ce qu'un équilibre de Nash soit détecté. Pour mWSLS et RWM, il n'y a pas de communication entre les joueurs et par conséquent, ils ne nécessitent aucun échange d'informations. Dans le tableau 2.1, nous résumons la complexité asymptotique et la quantité d'informations

échangées de chaque algorithme. Ce tableau montre l'efficacité de mWSLS et RWM en termes de complexité et d'échange d'informations. Dans les simulations dans la section suivante, nous verrons que mWSLS atteint des meilleures performances par rapport à RWM en termes de nombre d'utilisateurs servis, ce qui illustre la supériorité de notre algorithme proposé.

Algorithme / Mesure	mWSLS (par joueur)	RWM (par joueur)	max-SINR	BRD
Complexité	$\mathcal{O}(kT_{mWSLS})$	$\mathcal{O}(T_{RWM})$	$\mathcal{O}(knm)$	$\mathcal{O}(knmT_{min})$
Informations échangées	0	0	$\mathcal{O}(nm)$	$\mathcal{O}(T_{min})$

Tableau 2.1 Complexité et informations échangées des algorithmes distribués pour MSA.

2.9 Résultats de simulations

2.9.1 Résultats pour les algorithmes centralisés

Dans cette section, des simulations Monte Carlo sont effectuées pour illustrer les performances de ALG-MSA ainsi que l'étroitesse de la borne supérieure proposée. Les stations de base sont situées aléatoirement dans un carré de côté 100 mètres. Les gains des canaux sans fil sont modélisés comme dans (Zhuang *et al.*, 2016) et ils sont donnés par $15, 3 + 37, 6 \log_{10} d_{0 \rightarrow u}$ pour l'utilisateur u et une macro-cellule 0 et par $30, 6 + 36, 7 \log_{10} d_{b \rightarrow u}$ pour un utilisateur u et une petite-cellule b . Les solutions optimales de MSA et de SSA sont obtenues en résolvant les programmes mathématiques en nombres entiers (2.3) pour m quelconque et pour $m = 1$ respectivement. Sauf indication contraire, la puissance de transmission des utilisateurs est fixée à $p_u = 10$ dB, le seuil SINR requis pour chaque utilisateur est fixé à $\beta_u = 0$ dB, le nombre de *slots* est fixé à $m = 4$, le nombre d'utilisateurs est fixé à $k = 10$ et le nombre de stations de base est fixé à $n = 4$.

Les simulations Monte Carlo sont faites sur 1000 réalisations des canaux sans fil où dans chaque réalisation nous calculons les solutions des deux algorithmes. À la fin de la

simulation, nous calculons les valeurs moyennes correspondantes. Nous utilisons deux mesures de performance, à savoir le nombre moyen d'utilisateurs associés et l'efficacité spectrale moyenne, où l'efficacité spectrale est donnée par :

$$\sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}} \sum_{b \in \mathcal{B}} \log_2(1 + \text{SINR}_{u \rightarrow b}^s(\mathbf{X})),$$

où $\mathbf{X}$ est la solution d'association retournée par un algorithme donné. Nous rappelons que l'objectif de MSA (et aussi de SSA) est de maximiser le nombre d'utilisateurs associés. Par conséquent, OPT-MSA devrait toujours associer plus d'utilisateurs que ALG-MSA. Autrement dit, le nombre moyen d'utilisateurs associés obtenu par OPT-MSA sera toujours supérieur à celui obtenu par ALG-MSA. Cependant, comme l'objectif de MSA n'est pas d'optimiser l'efficacité spectrale, OPT-MSA peut donner une efficacité spectrale inférieure à celle donnée par ALG-MSA; tout dépend de la réalisation du problème. Heureusement, comme nous pouvons voir prochainement, l'algorithme proposé dans cette thèse ALG-MSA offre des meilleures performances en termes d'efficacité spectrale moyenne.

La figure 2.1 illustre le nombre moyen d'utilisateurs associés pour OPT-MSA et ALG-MSA en faisant varier m . Il est clair que la valeur de la solution retournée par ALG-MSA (A) est très proche de celle retournée par OPT-MSA (O) même lorsque m augmente (le gap est inférieur beaucoup à la borne supérieure de 2).

La figure 2.2 illustre le nombre moyen d'utilisateurs associés pour OPT-MSA et ALG-MSA en faisant varier n . L'écart entre O et A est petit et diminue davantage lorsque n augmente. Nous remarquons que les courbes convergent vers le nombre maximal d'utilisateurs dans le réseau, soit $k = 10$.

La figure 2.3 compare OPT-MSA et ALG-MSA en termes d'efficacité spectrale moyenne. Les deux algorithmes sont très proches l'un de l'autre bien que ALG-MSA donne une efficacité spectrale moyenne légèrement supérieure. Les figures 2.1 et 2.3 montrent que

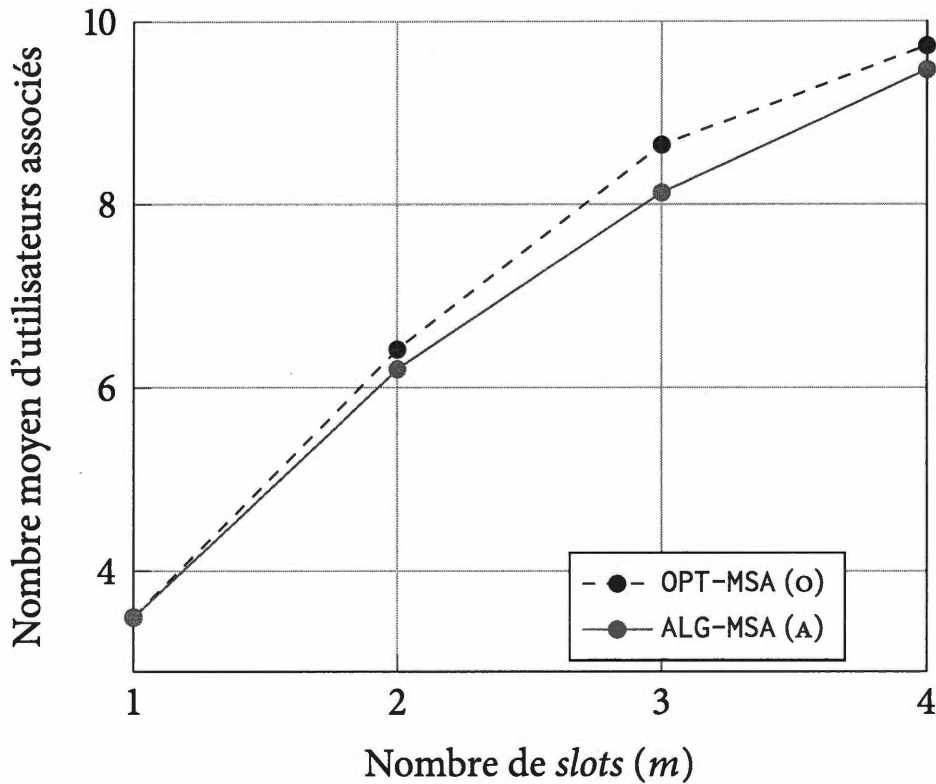


Figure 2.1 Nombre moyen d'utilisateurs associés par OPT-MSA et ALG-MSA en fonction de m pour MSA.

ALG-MSA permet d'atteindre un bon compromis entre l'association de plus d'utilisateurs et l'augmentation de l'efficacité spectrale. Nous concluons que maximiser le nombre d'utilisateurs associés permet de maximiser l'efficacité spectrale du réseau (c.-à-d. la capacité du réseau).

La figure 2.4 trace la fonction de répartition complémentaire de l'efficacité spectrale. Cette fonction est définie comme $F_{ES}(x) \triangleq \mathbf{P}[ES > x]$ où ES représente l'efficacité spectrale et $\mathbf{P}[A]$ représente la probabilité de l'évènement A . Nous remarquons que ALG-MSA atteint une meilleure probabilité par rapport à OPT-MSA. En effet, ALG-MSA réalise un gain en efficacité spectrale de presque 1,15 fois par rapport à OPT-MSA. Autrement dit, la

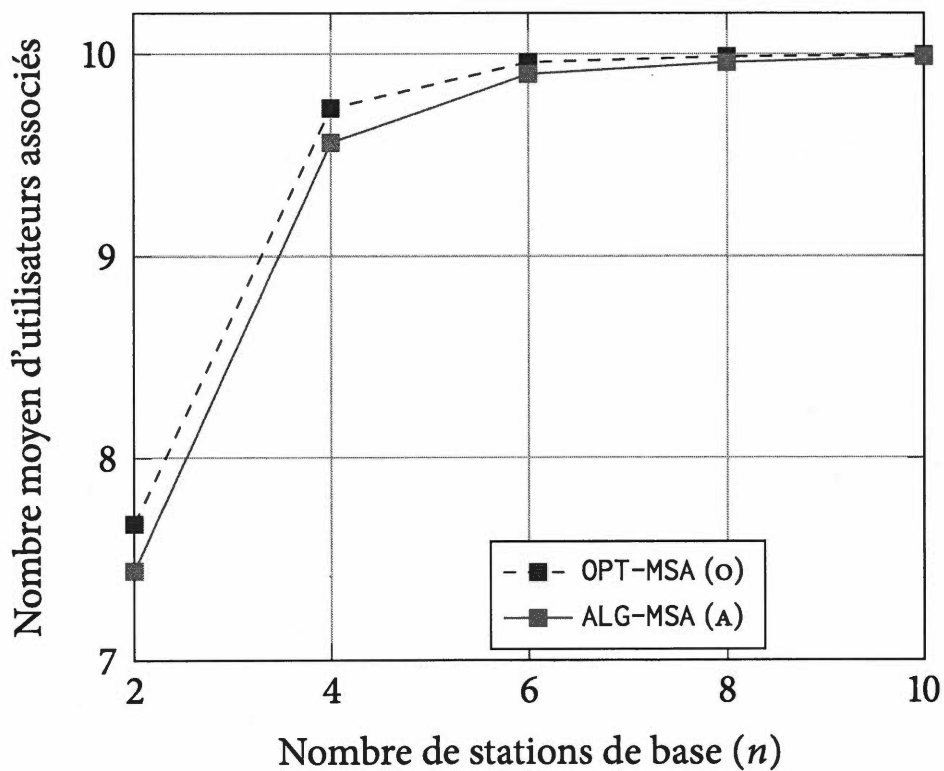


Figure 2.2 Nombre moyen d'utilisateurs associés par ALG-MSA et OPT-MSA en fonction de n pour MSA.

plus grande distance horizontale entre les deux courbes de la figure 2.4 est presque égale à 1,15. Par exemple, pour une probabilité de 0,8, l'efficacité spectrale de ALG-MSA atteint à peu près le 26,43 bits/s/Hz alors que celle de OPT-MSA atteint le 23,6 bits/s/Hz, soit un ratio de $26,43/23,6 = 1,12$.

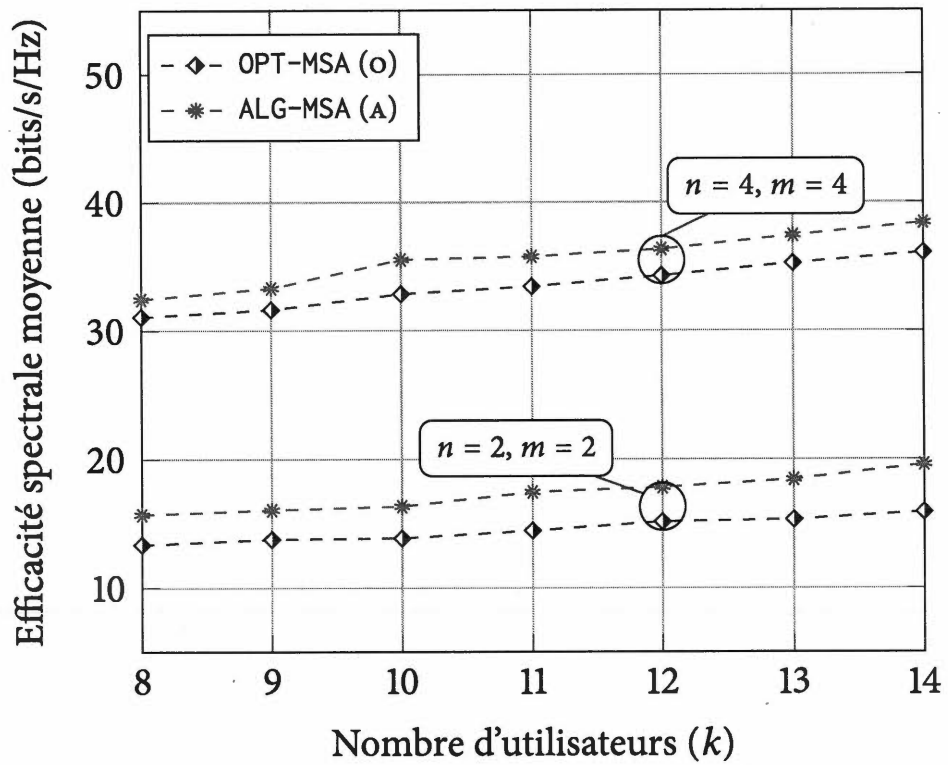


Figure 2.3 Efficacité spectrale moyenne par OPT-MSA et ALG-MSA en fonction de k pour MSA.

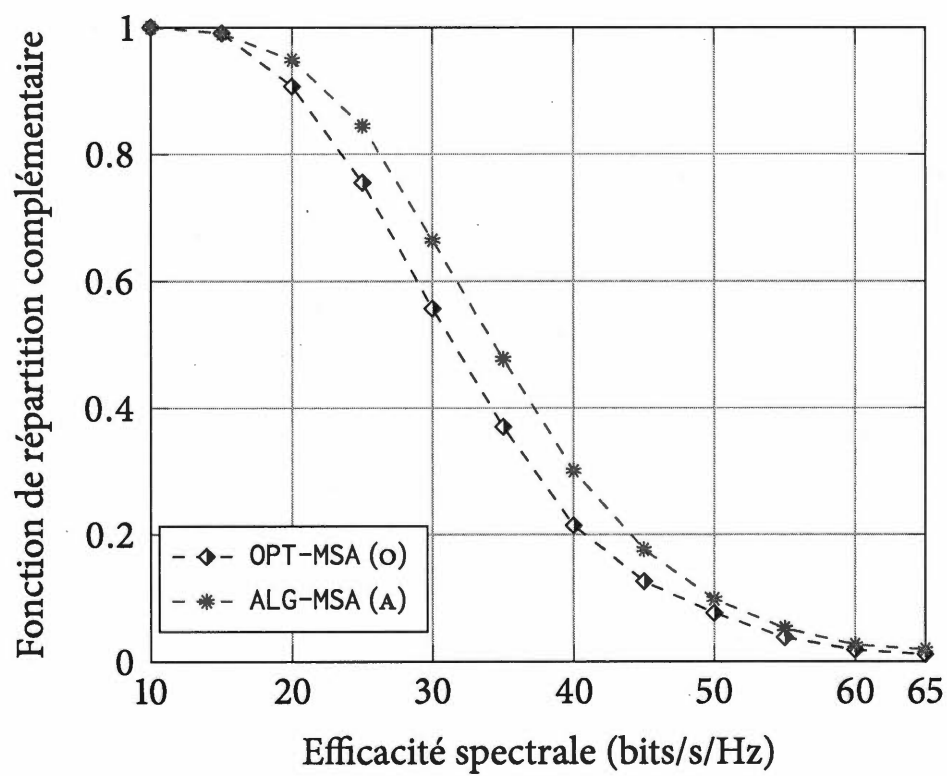


Figure 2.4 Fonction de répartition complémentaire de l'efficacité spectrale par OPT-MSA et ALG-MSA pour MSA.

2.9.2 Résultats pour les algorithmes distribués

Dans cette section, nous évaluons les performances des algorithmes distribués. Sauf indication contraire, nous utilisons les paramètres de simulation donnés dans le tableau 2.2.

Nom	Description	Valeur
β_b	SINR seuil	0 dB
p_0	Puissance de transmission de la macro-cellule	20 dB
p_b	Puissance de transmission de la petite-cellule	10 dB
k	Nombre d'utilisateurs	10
T_{mWSLS}	Nombre d'itérations de mWSLS	100
T_{RWM}	Nombre d'itérations de RWM	500
m	Nombre de slots	1

Tableau 2.2 Paramètres de simulations pour les algorithmes distribués pour MSA.

La figure 2.5 montre qu'il existe une valeur optimale $\tau^* = 0,1$ du facteur gagnant pour laquelle la performance de mWSLS est maximale. Cette valeur optimale est presque la même pour des valeurs différentes de n . En outre, lorsque n est grand, les petites valeurs de τ empêchent les joueurs de converger vers leurs actions gagnantes et par conséquent la performance de mWSLS diminue. Lorsque τ augmente, les joueurs n'explorent pas des nouvelles actions, mais plutôt exploitent les actions jouées et donc ils convergent rapidement ce qui diminue la performance de mWSLS.

La figure 2.6 montre qu'il existe une valeur optimale $\epsilon^* = 0,01$ du facteur perdant pour laquelle mWSLS associe le nombre maximum d'utilisateurs. Nous remarquons que lorsque n devient plus grand, des petites valeurs de ϵ dégradent la performance de mWSLS car elles empêchent les joueurs de choisir l'action de silence.

La figure 2.7 montre que la valeur optimale de τ entraîne des meilleures performances, mais a besoin de beaucoup d'itérations pour converger, alors que lorsque τ aug-

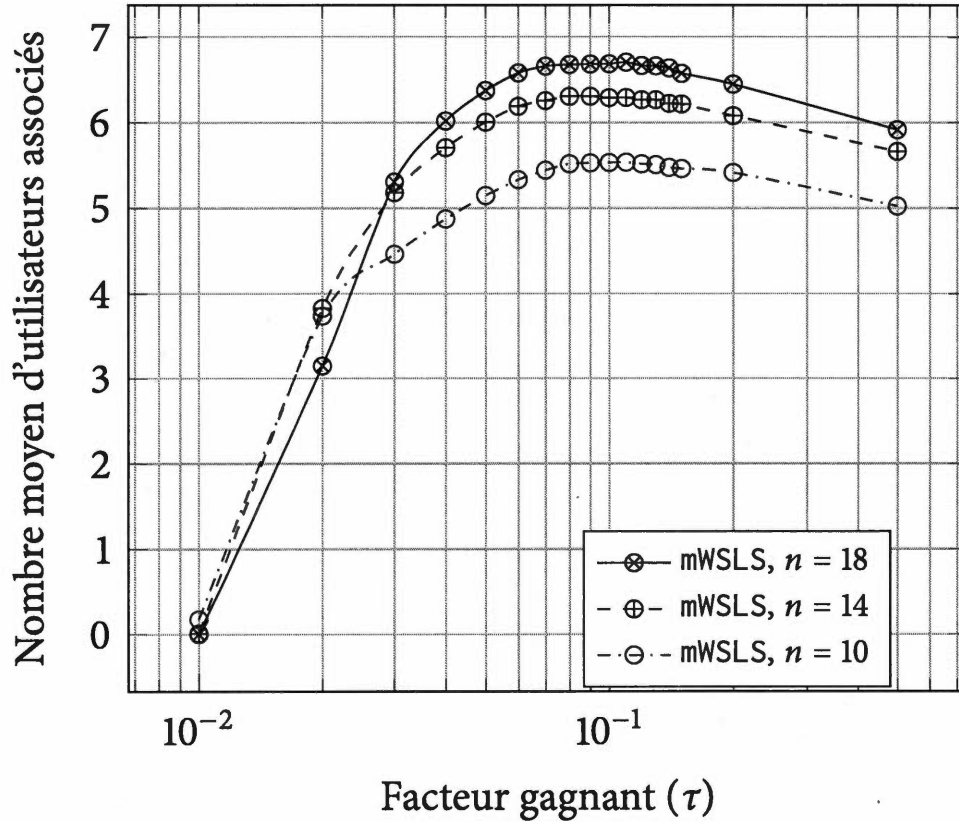


Figure 2.5 L'impact du facteur gagnant τ sur l'algorithme mWSLS pour MSA.

mente, mWSLS converge rapidement mais donne des performances faibles. Par exemple, pour $\tau = 0,3$, mWSLS associe trois utilisateurs en moins de 15 itérations alors que la même performance est obtenue après plus de 100 itérations avec $\tau = \tau^*$. Par conséquent, le facteur gagnant τ peut être considéré comme un paramètre de réglage pour ajuster un compromis entre la performance et le temps d'exécution. Nous pouvons conclure aussi que lorsque T_{mWSLS} devient plus grand, le délai augmente mais mWSLS maximise le nombre d'utilisateurs associés. Cependant, lorsque T_{mWSLS} devient plus petit, le délai devient faible et mWSLS termine rapidement mais la performance devient médiocre.

La figure 2.8 montre que mWSLS et BRD sont proches de l'optimal, même pour des grandes valeurs de n . Cependant, max-SINR et RWM donnent de mauvaises performances

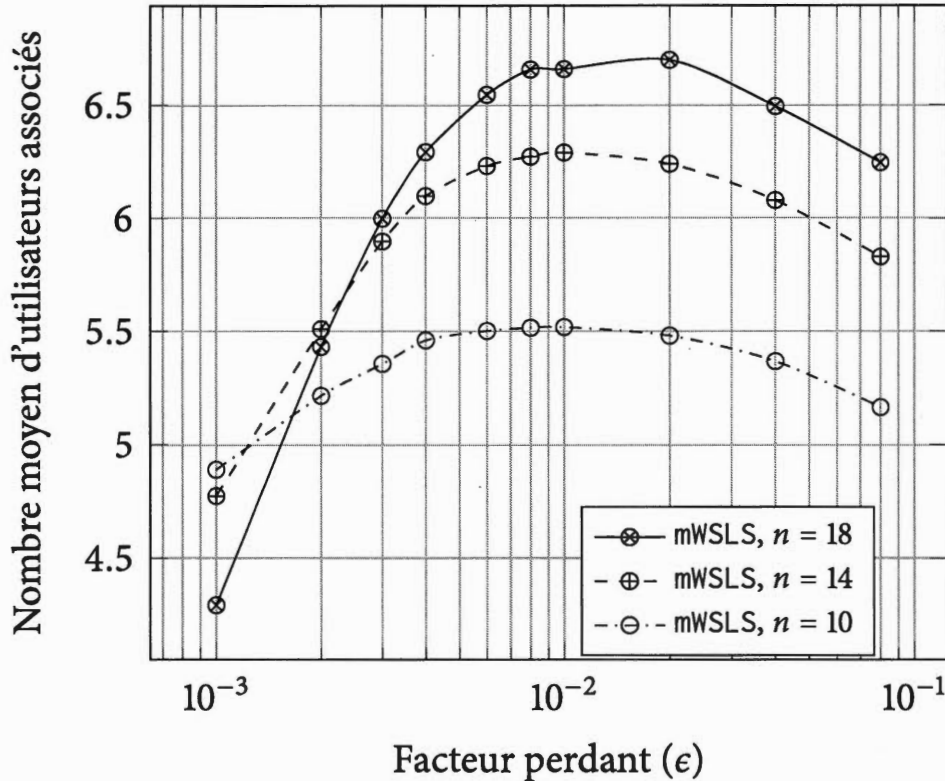


Figure 2.6 L'impact du facteur perdant ϵ sur l'algorithme mWSLS pour MSA.

lorsque n augmente. Malgré la supériorité de BRD par rapport à mWSLS, nous constatons que l'écart de performance entre eux reste presque identique même lorsque n devient plus grand. max-SINR donne des mauvaises performances car il surestime l'interférence dans le réseau puisqu'il suppose que chaque station de base transmet même si aucun utilisateur n'y est associé. Il est important de noter que même si RWM a une meilleure complexité asymptotique par rapport à mWSLS comme indiqué dans le tableau 2.1, il nécessite un nombre élevé d'itérations T_{RWM} pour atteindre des bonnes performances. Par exemple, pour $T_{RWM} = 500$ et $T_{mWSLS} = 100$, mWSLS surpasse RWM même pour $k = 10$.

Dans ce qui suit, nous montrons la performance de mWSLS en termes d'équilibrage de charge des stations de base. La mesure de la charge des stations de base est calculée en

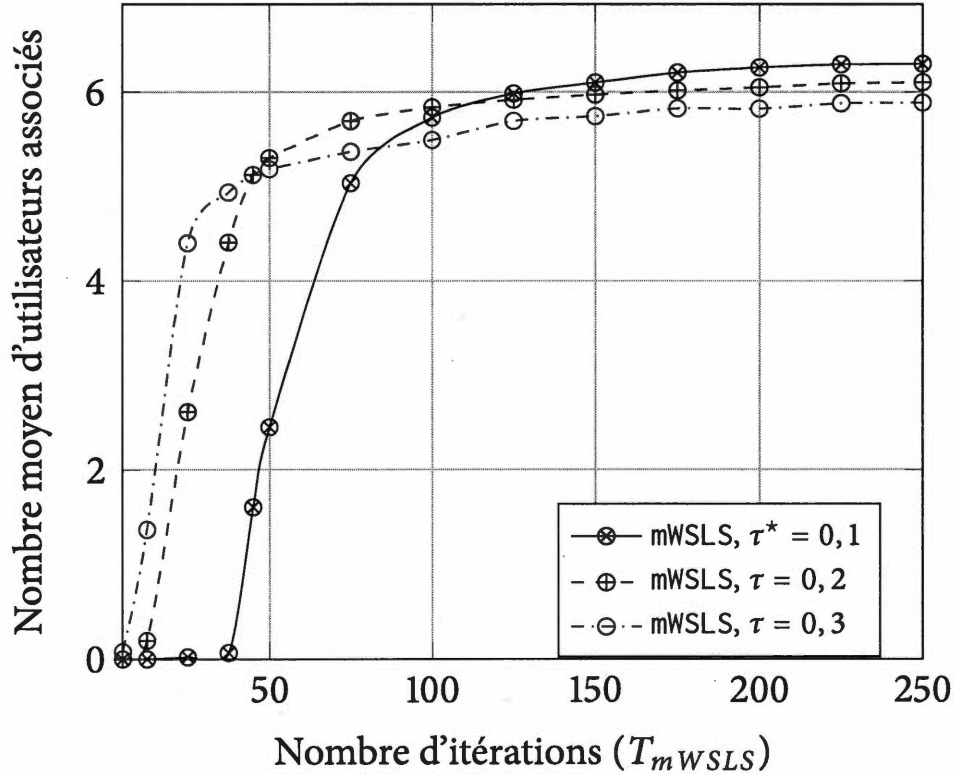


Figure 2.7 L'impact du nombre d'itérations sur l'algorithme mWSLS avec $\epsilon = \epsilon^*$ pour MSA.

utilisant l'indice de Jain (Jain *et al.*, 1984) comme suit :

$$\frac{(\sum_i \Omega_i)^2}{nm \sum_i \Omega_i^2},$$

où $\Omega \triangleq [\Omega_1, \Omega_2, \dots, \Omega_{nm}]$ est la charge des stations de base. La charge d'une station de base est le nombre d'utilisateurs qui y sont associés sur un ensemble de réalisations. Autrement dit, Ω_i représente le nombre d'utilisateurs associés au joueur i sur un ensemble de réalisations. Lorsque tous les joueurs associent le même nombre d'utilisateurs, $\Omega_1 = \Omega_2 = \dots = \Omega_{nm}$, alors l'indice de Jain est maximal.

Dans la figure 2.9, nous illustrons les performances de mWSLS et de sa version adaptée LB-mWSLS. LB-mWSLS fonctionne comme suit. Nous considérons une fenêtre de plu-

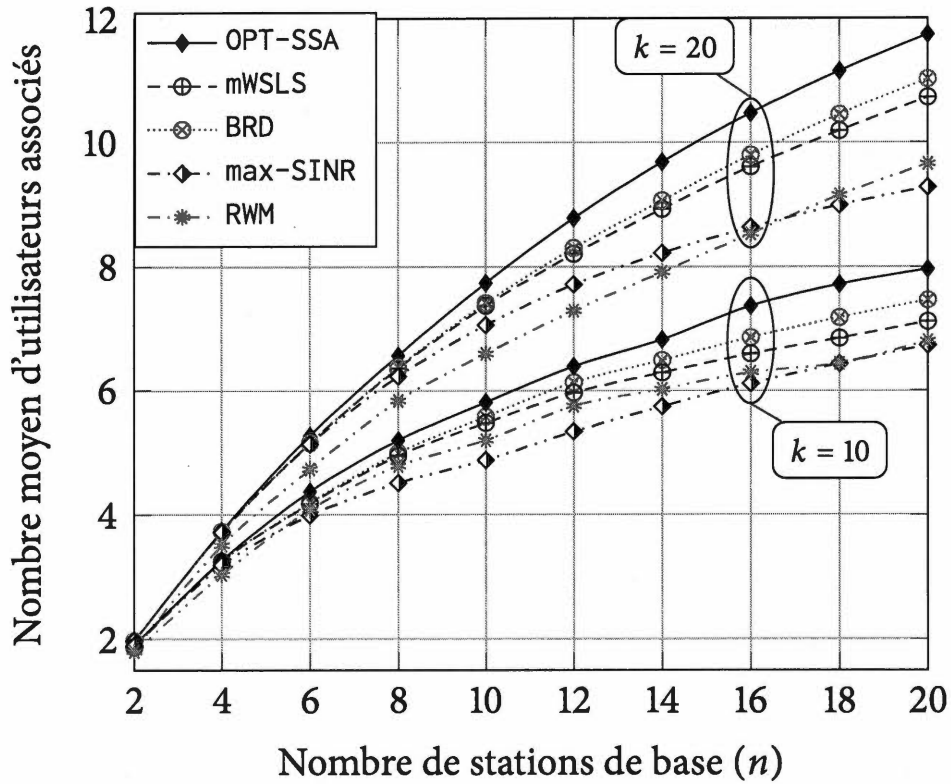


Figure 2.8 Nombre moyen d'utilisateurs associés par les algorithmes distribués en fonction de n pour MSA.

sieurs périodes pendant lesquelles mWSLS est exécuté plusieurs fois. L'idée de LB-mWSLS est de rendre la mise à jour des probabilités proportionnelle au nombre d'associations de chaque station de base au cours des périodes passées. Plus précisément, quand un joueur obtient une récompense égale à δ_1 , la mise à jour des probabilités ne change pas. D'autre part, lorsque sa récompense est égale à δ_3 , il doit augmenter la probabilité de jouer le silence proportionnellement à son histoire d'association, c'est-à-dire

$$\pi_{(b,s)}^0(t+1) = \pi_{(b,s)}^0(t) + \epsilon \Omega_{(b,s)} / nm, \quad (2.21)$$

$$\pi_{(b,s)}^u(t+1) = \pi_{(b,s)}^u(t) - \epsilon \Omega_{(b,s)} / nm, \quad \forall u \neq 0 \quad (2.22)$$

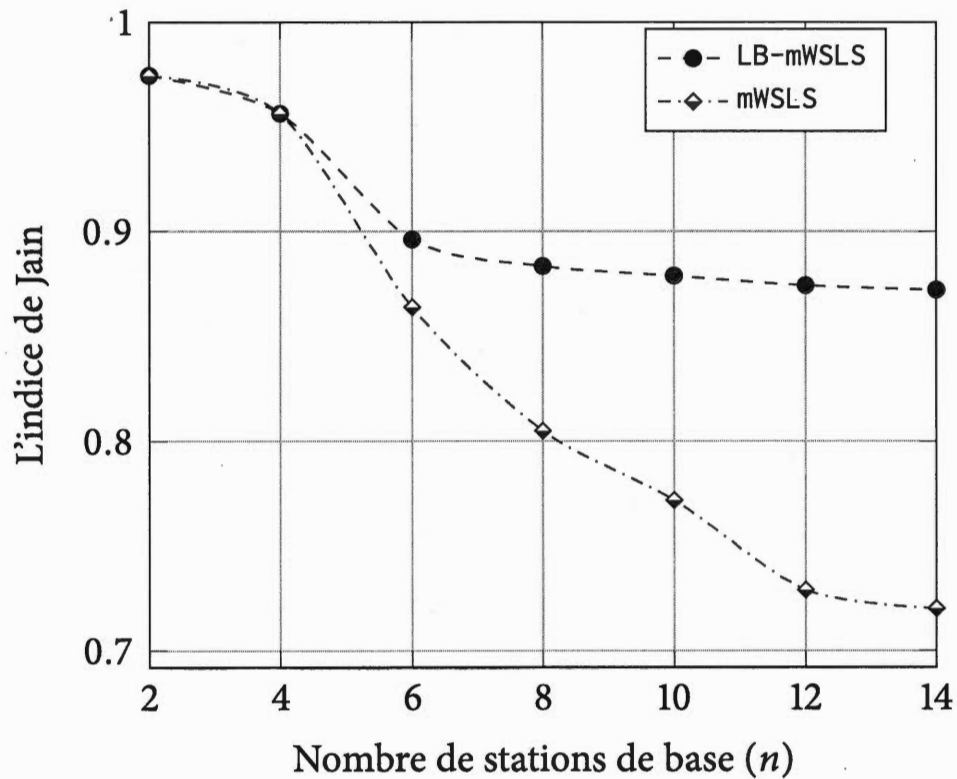


Figure 2.9 Équilibrage de charge pour les algorithmes mWSLS et LB-mWSLS en fonction de n pour MSA.

où $\Omega_{(b,s)}$ représente le nombre de périodes pendant lesquelles le joueur (b, s) est associé à un utilisateur. Par conséquent, un joueur qui n'est pas associé dans les périodes passées augmente plus lentement sa probabilité de silence que les autres joueurs. Ainsi, il est plus probable d'être associé dans la période actuelle.

Comme indiqué sur la figure 2.9, mWSLS n'équilibre pas bien la charge des stations de base surtout lorsque n augmente, p. ex. l'indice de Jain va aussi bas que 72% pour $n = 14$. Cependant, LB-mWSLS améliore la charge des stations de base et il atteint presque un indice de Jain égal à 87% pour tout $n > 6$.

Dans cette dernière partie, nous discutons les performances de $\mathcal{G}$ en termes du prix

d'anarchie (PoA) et prix de stabilité (PoS). Nous évaluons ces deux ratios en utilisant l'outil Gambit (McKelvey *et al.*, 2014) où les équilibres de Nash sont trouvés en utilisant une recherche exhaustive. Ces deux ratios sont définis comme suit (Koutsoupas et Papadimitriou, 1999; Agussurja et Lau, 2009). PoA est le ratio du pire équilibre de Nash sur la solution optimale alors que PoS est le ratio du meilleur équilibre de Nash sur la solution optimale.

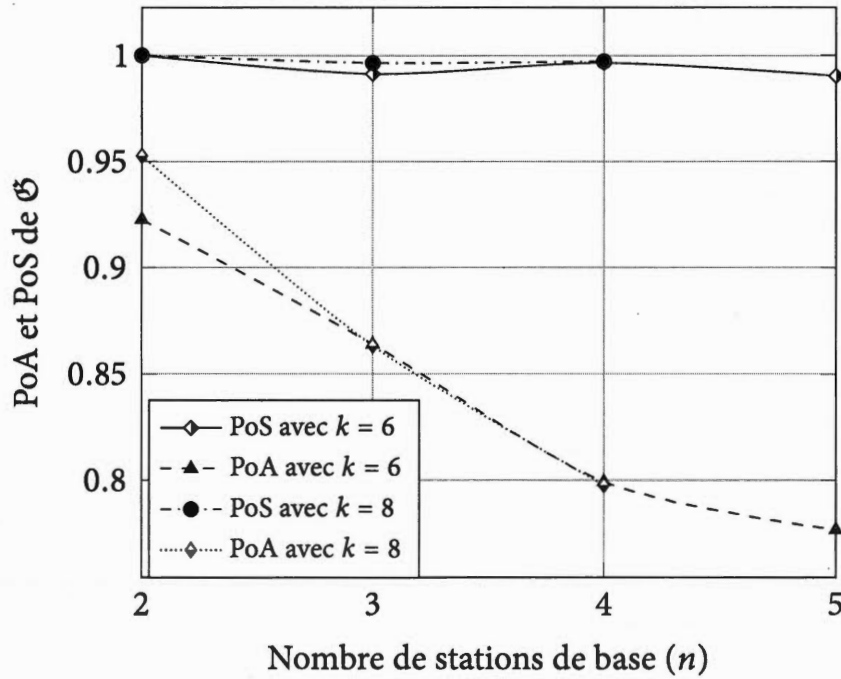


Figure 2.10 Le prix d'anarchie et de stabilité du jeu $\mathcal{G}$ en fonction de n pour MSA.

La figure 2.10 montre que les PoA et PoS de $\mathcal{G}$ sont proches de 1 surtout quand le nombre de joueurs est petit. Cependant, PoA se détériore à mesure que le nombre de joueurs augmente. Heureusement, PoS est toujours proche de 1. Ceci montre que l'ensemble des meilleurs équilibres de Nash de $\mathcal{G}$ est presque égal à l'ensemble optimal. Par conséquent, si l'équilibre de Nash est bien sélectionné (s'il existe), alors il doit être proche de l'optimal.

2.10 Conclusions

Dans ce chapitre, le problème de maximisation d'utilisateurs associés aux stations de base pendant un ensemble de *slots* et sous les contraintes de SINR est résolu dans un réseau DCN. Nous avons étudié la NP-difficulté de ce problème. Cette démonstration utilisait une réduction simple qui va permettre d'étudier la NP-difficulté d'autres problèmes connexes. Nous avons proposé un algorithme glouton qui permet de résoudre ce problème en l'étudiant dans un cas plus simple, soit celui d'un seul slot. Notre algorithme permet d'atteindre une borne supérieure étroite sur la valeur de la solution optimale. En outre, il permet d'atteindre une haute efficacité spectrale.

D'autre part, afin de réduire la dépendance à un point central qui constitue un goulot d'étranglement, nous avons modélisé et résolu ce problème de manière distribuée en utilisant la théorie des jeux non coopératifs. Tout d'abord, nous avons étudié les équilibres de Nash du jeu proposé. Ensuite, nous avons proposé un algorithme totalement distribué, mWSLS, qui ne suppose aucune coordination entre les stations de base et ne nécessite qu'un seul bit d'information de retour. Enfin, nous avons comparé mWSLS à (i) l'algorithme du meilleure réponse BRD, (ii) l'algorithme max-SINR, (iii) l'algorithme *Randomize Weighted Majority* (RWM), et (iv) l'algorithme optimal centralisé. Nous avons montré que mWSLS atteint des performances proches de l'optimal en termes de nombre moyen d'utilisateurs associés et nous avons montré aussi qu'une version modifiée de mWSLS peut efficacement équilibrer la charge du réseau.

CHAPITRE III

ACTIVATION DES STATIONS DE BASE ET ASSOCIATION DES UTILISATEURS POUR UNE MEILLEURE EFFICACITÉ ÉNERGÉTIQUE

Dans ce chapitre, nous étudions le problème d'association des utilisateurs afin d'améliorer l'efficacité énergétique en activant le minimum nombre de stations de base.

Le travail présenté dans ce chapitre a été publié dans l'article de revue (Mlika *et al.*, 2018a). Des résultats préliminaires de ce travail ont été publiés dans (Mlika *et al.*, 2017c).

3.1 Résumé

Ce chapitre étudie le problème conjoint d'activation des stations de base et d'association des utilisateurs dans les réseaux DCNs (appelé *Base station Operation and Association (BOA)* en anglais). L'objectif de BOA est de trouver l'ensemble optimal des stations de base à activer afin de leur associer tous les utilisateurs du réseau en garantissant un débit minimum pour chaque utilisateur. BOA est formulé comme un programme mathématique en nombres entiers non-convexe. Afin de le résoudre efficacement, nous distinguons deux cas : (i) débits minima élevés (dans ce cas, BOA est nommé BOAH) et (ii) débits minima faibles (dans ce cas, BOA est nommé BOAL). D'abord, nous montrons que la version de décision de BOAH est NP-complet, et après, nous réduisons ce problème conjoint à un problème d'activation de stations de base seulement. Cette réduction nous

a permis de développer un algorithme de force brute simple bien qu'il soit toujours un algorithme non-polynomial. Par conséquent, pour résoudre BOAH d'une manière efficace, nous proposons un algorithme polynomial heuristique. Quant à BOAL, puisque l'activation des stations de base et l'association des utilisateurs ne peuvent pas être découplées, nous simplifions le problème en supposant le pire cas d'interférence. Ainsi, BOAL est reformulé comme un programme linéaire en nombres entiers. Cependant, nous montrons que la version de décision de BOAL est NP-complet. Ainsi, nous proposons un algorithme glouton qui se caractérise par : (i) un facteur d'approximation logarithmique lorsqu'il trouve une solution faisable et (ii) un facteur d'approximation constant sinon. Finalement, des résultats de simulations Monte Carlo sont présentés pour illustrer les performances des algorithmes proposés.

3.2 Introduction

L'association des utilisateurs aux stations de base en maximisant le nombre d'utilisateurs associés permet d'atteindre une capacité élevée dans les réseaux DCNs comme nous avons vu dans le chapitre précédent. Cette association oblige certainement un grand nombre de stations de base à être activées afin de pouvoir associer beaucoup d'utilisateurs. L'activation d'un nombre élevé de stations de base engendre une consommation énergétique importante. En effet, la consommation énergétique qui correspond à la technologie de l'information et de la communication augmente avec un taux stupéfiant (Feng *et al.*, 2013). En particulier, les opérateurs de téléphonie mobile sont parmi les plus grands consommateurs d'énergie en raison de quantités énormes d'énergie consommées par les stations de base. En fait, environ 60% à 80% de l'énergie totale des réseaux cellulaires est consommée par les stations de base (Feng *et al.*, 2013; Rao et Fapojuwo, 2014). Encore pire, les stations de base consomment généralement de l'énergie même avec peu d'activité (Li *et al.*, 2016; Son *et al.*, 2011).

Par conséquent, le problème de maximisation de la capacité des réseaux DCNs ainsi que celui de minimisation de la consommation énergétique doit être résolu de manière conjointe.

Dans le présent chapitre, nous abordons les deux problèmes susmentionnés par le biais de l'association des utilisateurs et l'activation des stations de base. Dans ce chapitre, l'activation d'une station de base signifie de la mettre en marche (ou la mettre ON) et sa désactivation signifie de l'éteindre (ou la mettre OFF). Dans ce qui suit, nous utilisons le terme « ON/OFF » au lieu du terme « activation ». D'une part, le ON/OFF des stations de base peut réduire parfaitement l'énergie consommée en mettant en marche seulement les stations de base nécessaires. D'autre part, l'association des utilisateurs est l'une des méthodes les plus efficaces qui permet de réduire l'interférence dans les réseaux DCNs. Cependant, lorsque certaines stations de base sont ON ou OFF, la décision d'association des utilisateurs sera clairement affectée. Donc, le problème ON/OFF des stations de base et celui d'association des utilisateurs doivent être résolus conjointement afin d'obtenir une meilleure capacité ainsi qu'une meilleure efficacité énergétique. Ce problème conjoint est nommé comme *Base station Operation and Association (BOA)*.

3.2.1 Contributions

Dans ce chapitre, le problème conjoint d'ON/OFF des stations de base et d'association des utilisateurs (BOA) est étudié. BOA a pour objectif général de maximiser l'efficacité énergétique. Cet objectif est atteint en activant le plus petit nombre de stations de base tout en associant tous les utilisateurs et en respectant les exigences des utilisateurs en termes des débits minima. Afin de résoudre BOA, nous distinguons deux cas : (i) BOA avec débits minima élevés (BOAH) où le débit minimum requis pour chaque utilisateur est supérieur à 1 bits/s/Hz et (ii) BOA avec débits minima faibles (BOAL) où le débit minimum requis pour chaque utilisateur est inférieur à 1 bits/s/Hz.

Les contributions de cette partie de thèse sont résumées comme suit.

1. Nous formulons BOA comme un programme mathématique en nombres entiers non-convexe.
2. Nous prouvons que décider si une instance de BOAH est faisable ou pas est NP-difficile.
3. Nous prouvons que BOAH peut être réduit à un problème d'ON/OFF des stations de base seulement. Autrement dit, l'association des utilisateurs appartiendrait à la classe P dès que nous connaissons l'ensemble des stations de base ON.
4. En se basant sur cette réduction, nous présentons d'abord un algorithme de force brute simple. Deuxièmement, nous proposons un algorithme heuristique efficace qui trouve des solutions faisables très souvent. Troisièmement, nous formulons BOAH comme un programme linéaire en nombres entiers.
5. Quant à BOAL, puisqu'il ne peut pas être découplé comme BOAH, nous le simplifions en supposant le pire cas d'interférence comme dans (Pollakis *et al.*, 2012; Zhuang *et al.*, 2016). Ensuite, nous formulons BOAL comme un programme linéaire en nombres entiers. Bien que l'hypothèse du pire cas d'interférence simplifie le problème, nous montrons que BOAL reste NP-difficile.
6. Nous concevons un algorithme d'approximation pour résoudre BOAL. Cet algorithme donne (i) un facteur d'approximation logarithmique lorsqu'il trouve des solutions faisables et (ii) un facteur d'approximation constant sinon.
7. Nous effectuons des simulations Monte Carlo afin d'illustrer les performances des algorithmes proposés.

3.2.2 Organisation

Le reste du chapitre est organisé comme suit. La section 3.3 fait une revue de littérature pertinente. La section 3.4 présente le modèle du système et formule BOA comme un

programme mathématique non-linéaire. Ensuite, la section 3.5 présente BOAH, montre qu'il s'agit d'un problème NP-difficile et présente les algorithmes proposés pour le résoudre. Dans la section 3.6, nous simplifions BOAL, étudions sa NP-difficulté et discutons les algorithmes proposés pour le résoudre. La section 3.7 présente les résultats de simulations et la section 3.8 conclut ce chapitre.

3.3 Revue de littérature

La plupart des travaux de recherche n'étudient pas BOA d'une manière conjointe. En outre, ils n'étudient pas sa complexité computationnelle (sa NP-difficulté). En plus, la majorité des solutions proposées dans la littérature pour BOA se basent sur des algorithmes heuristiques qui ne garantissent pas théoriquement les valeurs des solutions obtenues. Par conséquent, nous nous concentrons principalement sur (i) l'analyse de la complexité computationnelle du BOA et (ii) la conception d'algorithmes d'approximation avec des garanties théoriques dans le pire cas.

Les deux articles (Rao et Fapojuwo, 2014; Liu *et al.*, 2016) résument respectivement les études récentes du problème de l'optimisation de l'efficacité énergétique et celui de l'association des utilisateurs dans les réseaux DCNs. Dans ce qui suit nous citons les travaux de recherche les plus liés à notre travail.

Dans (Li *et al.*, 2016), les auteurs étudient le problème ON/OFF des stations de base, association des utilisateurs, allocation de fréquences et de puissance dans un réseau DCN. Le problème est formulé comme étant un programme mathématique stochastique avec l'objectif de minimiser la consommation totale de puissance. Les auteurs décomposent le problème en différents sous-problèmes et résolvent chaque sous-problème séparément. Leur technique se base sur la méthode d'optimisation de Lyapunov. Bien que les auteurs étudient BOA, ils ne se concentrent pas sur l'analyse de sa NP-difficulté ni sur le développement d'algorithmes d'approximation. Dans (Son *et al.*, 2011), le problème ON/OFF

des stations de base et association des utilisateurs est étudié afin d'atteindre un bon compromis entre le délai et l'efficacité énergétique. Les auteurs décomposent le problème en deux sous-problèmes : (i) association d'utilisateurs, et (ii) ON/OFF des stations de base. Pour le premier sous-problème, ils proposent un algorithme optimal et présentent une implémentation distribuée. Pour le second sous-problème, ils affirment qu'il s'agit d'un problème combinatoire et donc difficile à résoudre sans donner une preuve formelle. En se basant sur cette affirmation, ils proposent un algorithme heuristique simple inspiré de la maximisation d'une fonction sous-modulaire. En plus de ne pas étudier sa NP-difficulté et ne pas proposer des algorithmes d'approximation (Li *et al.*, 2016; Son *et al.*, 2011) supposent un modèle de système pour BOA principalement différent de notre modèle de système. Dans (Pollakis *et al.*, 2012), le problème ON/OFF des stations de base et association des utilisateurs dans les réseaux DCNs est étudié où l'allocation universelle de fréquence est supposée. Les auteurs visent à minimiser le nombre de stations de base ON en garantissant une qualité de service minimale aux utilisateurs. Ils proposent une méthode basée sur l'algorithme de « *majorize-minimization (MM)* ». Dans (Zhuang *et al.*, 2016), les auteurs étudient une version généralisée du problème étudié dans (Pollakis *et al.*, 2012) où l'allocation fractionnelle de fréquence est considérée. Les auteurs proposent un algorithme centralisé en utilisant la technique de minimisation du norme ℓ_1 pondérée. Dans (Feng *et al.*, 2016), les auteurs étudient le problème ON/OFF des stations de base dans les réseaux DCNs avec des antennes multiples. Ils utilisent la technique de programmation mathématique qui consiste à relaxer les variables binaires et ils forment leur problème comme un programme mathématique convexe en utilisant des techniques de décomposition. Les travaux (Pollakis *et al.*, 2012; Zhuang *et al.*, 2016; Feng *et al.*, 2016) ne fournissent pas des études de NP-difficulté et revendiquent la difficulté de leurs problèmes sans preuves formelles. En outre, ils ne proposent pas des algorithmes d'approximation qui garantissent des performances dans le pire cas. Dans (Ye *et al.*, 2016a) et (Lin *et al.*, 2016), afin d'améliorer l'efficacité énergétique, le problème d'asso-

ciation des utilisateurs et d'allocation de puissance dans les réseaux DCNs avec et sans des antennes multiples à grande échelle, respectivement, est étudié. Les auteurs revendiquent la difficulté de leurs problèmes sans fournir des preuves de NP-difficulté formelles. Les auteurs de (Ye *et al.*, 2016a) formulent leur problème comme un programme mathématique et utilisent la relaxation des variables binaires, puis proposent un algorithme itératif. D'autre part, les auteurs de (Lin *et al.*, 2016) proposent une méthode de décomposition. Dans (Tam *et al.*, 2017), le problème d'association des utilisateurs et d'allocation de puissance pour équilibrer la charge des réseaux DCNs est considéré. L'objectif est divisé en deux sous-objectifs : (i) maximiser le débit total et (ii) maximiser le débit minimum individuel. Les auteurs revendiquent que leur problème est NP-difficile. Ensuite, ils développent une méthode itérative qui résout l'association des utilisateurs et l'allocation de puissance séparément. Plus précisément, l'association des utilisateurs est résolue par une méthode de pénalité alors que l'allocation de puissance est résolue par une méthode de programmation convexe successive. Les méthodes développées ne garantissent pas théoriquement des performances par rapport aux solutions optimales. En plus de ne pas étudier la NP-difficulté et de ne pas proposer des algorithmes d'approximation pour BOA, les travaux de recherche cités ci-dessus ignorent un aspect très important de BOA ; soit sa faisabilité. Comme l'objectif de BOA est de minimiser le nombre de stations de base à activer en associant tous les utilisateurs et en garantissant leurs débits minima, il peut ne pas être faisable.

Dans ce chapitre, nous comblons les lacunes de recherche suivantes. D'abord, nous étudions la faisabilité de BOA, puis nous analysons sa NP-difficulté, et enfin nous proposons des algorithmes optimaux, heuristiques et d'approximation efficaces.

3.4 Modèle de système et formulation du problème

Nous considérons un réseau DCN composé de n petites-cellules, une macro-cellule et k utilisateurs. L'ensemble des stations de base est noté par $\mathcal{B} \triangleq \{0, 1, \dots, n\}$ où l'indice 0 désigne la macro-cellule alors que les indices $1, 2, \dots, n$ désignent les petites-cellules. L'ensemble des utilisateurs est noté par $\mathcal{U} \triangleq \{1, 2, \dots, k\}$. La transmission sur le lien descendant, où les stations de base envoient des messages aux utilisateurs, est considérée. Les petites-cellules peuvent être activées (ON) ou désactivées (OFF) alors que la macro-cellule est toujours ON. La puissance de transmission de la station de base b , quand elle est ON, est notée par p_b , et elle est supposée inclure un terme de la consommation de puissance statique. C'est-à-dire, la puissance de transmission de la station de base b s'exprime comme $p_b \triangleq p_b^r + p_b^s$ où p_b^r est sa puissance rayonnée alors que p_b^s est la puissance statique dissipée dans ses circuits. Chaque utilisateur demande un débit minimum R^{min} lorsqu'il est associé à une station de base quelconque. Nous supposons que chaque utilisateur peut être associé à une seule station de base, mais une station de base peut associer plusieurs utilisateurs qui partageraient ses ressources d'une façon équitable. Le problème *Base station Operation and Association (BOA)* répond à la question suivante : comment associer tous les utilisateurs aux stations de base de telle sorte que les exigences des débits minima des utilisateurs soient respectées et le nombre de stations de base ON soit aussi petit que possible ?

Avant de répondre à cette question, nous formulons BOA comme un programme mathématique en nombres entiers. Pour ce faire, les variables d'optimisation $x_{b \rightarrow u}$ et y_b respectives suivantes sont utilisées.

$$x_{b \rightarrow u} \triangleq \begin{cases} 1, & \text{si } u \text{ est associé à } b \\ 0, & \text{sinon.} \end{cases}, \quad (3.1)$$

$$y_b \triangleq \begin{cases} 1, & \text{si } b \neq 0 \text{ est ON} \\ 0, & \text{sinon.} \end{cases} \quad (3.2)$$

En notation matricielle, les variables $\mathbf{X}$ et $\mathbf{y}$ sont utilisées, c.-à-d., $\mathbf{X} \triangleq [x_{b \rightarrow u}]$ et $\mathbf{y} \triangleq [y_0, y_1, \dots, y_n]$.

Lorsque l'utilisateur u est associé à la station de base b , le rapport signal-sur-interférence-plus-bruit (SINR) est donné par :

$$\text{SINR}_{b \rightarrow u}(\mathbf{y}) = \frac{g_{b \rightarrow u}}{1 + \sum_{\substack{b' \in \mathcal{B} \\ b' \neq b}} g_{b' \rightarrow u} y_{b'}}, \quad (3.3)$$

où $g_{b \rightarrow u} \triangleq p_n^r h_{b \rightarrow u} / \sigma^2$ est le rapport signal-sur-bruit (SNR). Le terme σ^2 désigne la puissance du bruit additif blanc gaussien (AWGN). Le terme $h_{b \rightarrow u}$ est modélisé comme une variable aléatoire qui prend en compte l'évanouissement à grande échelle comme dans (Zhuang *et al.*, 2016). En général, $h_{b \rightarrow u}$ peut s'écrire comme (Goldsmith, 2005) :

$$h_{b \rightarrow u} = K_0 \left(\frac{d_0}{d_{b \rightarrow u}} \right)^\alpha, \quad (3.4)$$

avec α l'exposant d'affaiblissement, $d_{b \rightarrow u}$ la distance entre la station de base b et l'utilisateur u , d_0 est une distance de référence et K_0 est une constante sans unité qui dépend des caractéristiques de l'antenne et de l'atténuation moyenne du canal (Goldsmith, 2005).

Pour simplifier les notations et afin d'isoler les performances de la couche physique des détails spécifiques d'implémentation, nous utilisons la capacité de canal de Shannon comme une approximation du débit de transmission (Lau, 2005). Par conséquent, le débit de transmission entre la station de base b et l'utilisateur u , lorsque ce dernier y est associé, est donné par :

$$R_{b \rightarrow u}(\mathbf{y}) = \log_2 (1 + \text{SINR}_{b \rightarrow u}(\mathbf{y})), \text{ [en bits/s/Hz]} \quad (3.5)$$

et nous définissons le débit de transmission effectif (Tam *et al.*, 2017) entre la station de base b et l'utilisateur u comme :

$$R_{b \rightarrow u}^{eff}(\mathbf{X}, \mathbf{y}) \triangleq \frac{R_{b \rightarrow u}(\mathbf{y})}{\sum_{u' \in \mathcal{U}} x_{b \rightarrow u'}}, \text{ [en bits/s/Hz]} \quad (3.6)$$

où $\sum_{u' \in \mathcal{U}} x_{b \rightarrow u'}$ est le nombre d'utilisateurs associés à b .

En fait, $R_{b \rightarrow u}(\mathbf{y}) = R_{b \rightarrow u}^{eff}(\mathbf{X}, \mathbf{y})$ lorsqu'un seul utilisateur est associé à b . Par contre, lorsque plusieurs utilisateurs y sont associés, elle divise d'abord équitablement ses ressources temporelles (ou un autre type de ressources) en des *slots*. Ensuite, elle alloue uniformément ces *slots* à ses utilisateurs (Tam *et al.*, 2017). Comme ça il n'y aura pas d'interférence entre les utilisateurs de la même station de base.

L'hypothèse d'évanouissement à grande échelle utilisée dans ce travail permet de supposer que le canal sans fil ne change pas rapidement. Cette hypothèse est largement utilisée dans la littérature, p. ex. (Zhuang *et al.*, 2016; Wang *et al.*, 2016). En outre, si nous utilisons l'évanouissement à petite échelle (*small-scale fading* en anglais), alors nous devons utiliser respectivement le débit de transmission ergodique ainsi que le débit de transmission ergodique effectif au lieu de (3.5) et (3.6) comme indiqué dans (Zhuang *et al.*, 2016).

En utilisant les notations et définitions précédentes, BOA peut être formulé comme suit :

$$\text{minimiser} \quad \sum_{b=1}^n y_b \quad (3.7a)$$

$$\text{sous les contraintes} \quad x_{b \rightarrow u}, y_b \in \{0, 1\}, \forall b \in \mathcal{B}, u \in \mathcal{U}, \quad (3.7b)$$

$$\sum_{b \in \mathcal{B}} x_{b \rightarrow u} = 1, \forall u \in \mathcal{U}, \quad (3.7c)$$

$$x_{b \rightarrow u} \leq y_b, \forall b \in \mathcal{B}, u \in \mathcal{U}, \quad (3.7d)$$

$$x_{b \rightarrow u} R_{b \rightarrow u}^{eff}(\mathbf{X}, \mathbf{y}) \geq x_{b \rightarrow u} R^{min}, \forall b \in \mathcal{B}, u \in \mathcal{U}. \quad (3.7e)$$

La fonction objectif (3.7a) minimise le nombre de stations de base ON. Les contraintes (3.7b) garantissent que les variables $x_{b \rightarrow u}$ et y_b sont binaires pour tous b, u . Les contraintes (3.7c) garantissent que chaque utilisateur est associé à une seule station de base. Les contraintes (3.7d) garantissent que si une station de base est OFF, alors aucun utilisateur n'y est associé. Finalement, les contraintes (3.7e) garantissent le débit minimum R^{min} de chaque utilisateur associé. Le problème (3.7) est non-linéaire parce que les contraintes (3.7e) sont non-convexes.

Pour faciliter la présentation, nous reformulons les équations (3.3), (3.5) et (3.6) en utilisant des ensembles. Soit $\mathcal{S} \triangleq \{b \in \mathcal{B} : y_b = 1\}$ l'ensemble des stations de base ON. Ces trois équations peuvent être réécrites, respectivement, comme suit :

$$\text{SINR}_{b \rightarrow u}(\mathcal{S}) \triangleq \frac{g_{b \rightarrow u}}{1 + \sum_{\substack{b' \in \mathcal{S} \\ b' \neq b}} g_{b' \rightarrow u}},$$

$$R_{b \rightarrow u}(\mathcal{S}) \triangleq \log_2(1 + \text{SINR}_{b \rightarrow u}(\mathcal{S})),$$

et

$$R_{b \rightarrow u}^{eff}(\mathcal{S}) \triangleq \frac{R_{b \rightarrow u}(\mathcal{S})}{k_b},$$

où k_b représente le nombre d'utilisateurs associés à la station de base b .

Pour résoudre BOA, nous adoptons une approche centralisée. En d'autres termes, un point central connaît tous les gains des canaux sans fil entre les utilisateurs et les stations de base. Ces informations peuvent être collectées en estimant les distances entre les utilisateurs et les stations de base, p. ex., en mesurant l'intensité des signaux reçus (Chen et Feng, 2017), sans avoir besoin de l'emplacement exact ni des informations complètes sur l'état du canal.

Symbole	Description
k	Nombre d'utilisateurs
n	Nombre de stations de base
m	$\min(k, n)$
k_b	Nombre d'utilisateurs associés à b
$\mathcal{U}$	Ensemble des utilisateurs
$\mathcal{B}$	Ensemble des stations de base
$\mathcal{B} \setminus \mathcal{S}$	Différence d'ensembles, $\{x : x \in \mathcal{B} \text{ et } x \notin \mathcal{S}\}$
$\mathcal{U}_b$	Ensemble des utilisateurs associés à b
$\mathcal{S}, \mathcal{S}_1, \text{ et } \mathcal{S}_2$	Ensemble des stations de base ON
$p_b, p_b^r, \text{ et } p_b^s$	Puissance de transmission, rayonnée, et statique de b
$g_{b \rightarrow u}$	Le SNR entre b et u
$h_{b \rightarrow u}$	Le gain du canal sans fil entre b et u
σ^2	La puissance du bruit AWGN
$d_{b \rightarrow u}$	La distance entre b et u
$R^{\min}$	Le débit minimum requis
$R_{b \rightarrow u}$	Le débit de transmission entre b et u
$R_{b \rightarrow u}^{\text{eff}}$	Le débit de transmission effectif entre b et u
$\mathbf{X} = [x_{b \rightarrow u}]$	La matrice d'association
$\mathbf{y} = [y_1, \dots, y_n]$	Le vecteur d'ON/OFF
$\mathbf{X}_u$ (resp. $\mathbf{X}^b$)	La u -ème ligne (resp. b -ème colonne) de $\mathbf{X}$

Tableau 3.1 Liste des notations pour BOA.

Dans le reste de ce chapitre, nous utilisons les notations suivantes. Un ensemble est noté par une lettre majuscule calligraphique et sa cardinalité est notée par une lettre minuscule italique. Le coefficient binomial, qui donne le nombre de parties de k éléments dans un ensemble de n éléments, est noté par $\binom{n}{k}$. La plupart des notations utilisées sont données dans le tableau 3.1.

3.5 BOA avec débits minima élevés

Dans cette section, le débit minimum R^{min} est supposé être supérieur ou égal à 1 bits/s/Hz (c.-à-d., BOAH). Premièrement, nous démontrons que BOAH est NP-complet. Ensuite, nous réduisons BOAH à un problème d'ON/OFF des stations de base seulement. En se basant sur cette réduction, nous proposons un algorithme de force brute ainsi qu'un algorithme glouton efficace.

3.5.1 NP-difficulté

Nous montrons que même trouver une solution faisable à BOAH est NP-difficile. La preuve est basée sur une réduction polynomiale à partir du problème de couverture exacte par trois ensembles (en anglais, *Exact Cover by 3 sets (3EC)*)¹. Avant de commencer la démonstration, nous donnons quelques définitions.

Définition 3.5.1 (3EC (Garey et Johnson, 1979)) :

INSTANCE : un ensemble $\mathcal{X}$ de x entiers, appelé l'univers, où x s'écrit comme $x = 3r$ pour tout entier positif r , et une collection $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_c\}$ de c sous-ensembles de $\mathcal{X}$ où chaque ensemble $\mathcal{C}_i$ est de cardinalité 3, c.-à-d. $|\mathcal{C}_i| \triangleq c_i = 3$.

QUESTION : existe-t-il une sous-collection $\mathcal{C}^*$ de $\mathcal{C}$ de $x/3 (= r)$ sous-ensembles qui couvrent exactement $\mathcal{X}$? C'est-à-dire, les sous-ensembles de $\mathcal{C}^*$ sont disjoints deux à deux et que leur union est égale à $\mathcal{X}$.

Il est bien connu que 3EC est NP-complet (Garey et Johnson, 1979). La version de faisabilité de BOAH (notée FBOAH) est définie comme suit :

1. Il est à noter que le choix de 3EC se base sur le fait que le nombre d'utilisateurs pouvant être associés à n'importe quelle station de base est limité à 3.

Définition 3.5.2 (FBOAH) :

INSTANCE : un ensemble $\mathcal{B}$ de stations de base, un ensemble $\mathcal{U}$ d'utilisateurs, un entier positif $\beta \leq n$, un débit minimum R^{min} , et les rapports SNR $g_{b \rightarrow u}$ pour chaque station de base b et utilisateur u .

QUESTION : existe-t-il un sous-ensemble $\mathcal{S}$ de $\mathcal{B}$ de β stations de base et une collection $\{\mathcal{U}_b, b \in \mathcal{S}\}$ d'ensembles d'utilisateurs tels que $\mathcal{U}_b \cap \mathcal{U}_{b'} = \emptyset$ pour tout $b \neq b'$, $\bigcup_{b \in \mathcal{S}} \mathcal{U}_b = \mathcal{U}$ et $R_{b \rightarrow u}(\mathcal{S}) \geq R^{min} k_b$ pour tout $b \in \mathcal{S}$ et $u \in \mathcal{U}_b$ (où $k_b \triangleq |\mathcal{U}_b|$) ?

Intuitivement, FBOAH et 3EC peuvent être vus comme deux problèmes équivalents au sens suivant. Chaque ensemble de la collection $\mathcal{C}$ correspond à un ensemble d'utilisateurs choisis de manière à satisfaire aux débits minima. Nous aimerions couvrir tous les utilisateurs en choisissant un nombre donné d'ensembles d'utilisateurs. Le prochain théorème donne une preuve formelle de cette intuition.

Théorème 3.5.3 FBOAH est NP-complet.

Démonstration : soit $I = (\mathcal{X}, \mathcal{C})$ une instance de 3EC. Nous devons créer en temps polynomial *une* instance I' de FBOAH tel que, FBOAH est résolu avec I' si et seulement si 3EC est résolu avec I . Soit $I' = (\mathcal{B}, \mathcal{U}, \beta, R^{min}, [g_{b \rightarrow u}])$ l'instance de FBOAH suivante :

1. $\mathcal{U} = \mathcal{X}$, $n = c$, $\beta = r$ et R^{min} est choisi ²= 1 bits/s/Hz. Autrement dit, l'ensemble des utilisateurs représente l'univers $\mathcal{X}$ et donc le nombre d'utilisateurs k est égal à $3r$.

Le nombre de stations de base est $n = c$. Nous devons couvrir les $3r$ utilisateurs avec $\beta = r$ stations de base et avec un débit minimum égal à 1 bits/s/Hz.

2. Dans le théorème 3.5.3, R^{min} est choisi égal à 1 bits/s/Hz pour créer une instance de FBOAH. Il est à noter que nous pouvons créer une autre instance de FBOAH en choisissant n'importe quelle valeur de R^{min} qui doit être supérieure ou égale à 1 bits/s/Hz ; en modifiant la valeur des gains $g_{b \rightarrow u}$ en conséquence.

2. Soit $g_{b \rightarrow u}$ la matrice suivante :

$$g_{b \rightarrow u} \triangleq \begin{cases} \frac{7}{3}k, & \text{si } u \in \mathcal{C}_b \\ 1, & \text{sinon} \end{cases}$$

La valeur $7/3$ est choisie arbitrairement afin de satisfaire aux exigences des débits minima des utilisateurs comme nous le verrons bientôt. N'importe quelle valeur supérieure à $7/3$ reste toujours un choix valide.

L'instance I' est clairement créée en temps polynomial. Il reste maintenant à montrer que 3EC est résolu si et seulement si FBOAH est résolu.

— Pour le sens de *seulement si*, supposons que 3EC est résolu. En d'autres termes, il existe une sous-collection $\mathcal{C}^* = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_r\}$ de r ensembles tels que $\bigcup_{i=1}^r \mathcal{C}_i = \mathcal{X}$ et $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ pour tout $i \neq j$. Une solution faisable de FBOAH consiste à choisir $\mathcal{S} = \{1, 2, \dots, r\}$ comme l'ensemble des stations de base et $\mathcal{U}_b = \mathcal{C}_b$ comme l'ensemble des utilisateurs associés à la station de base $b \in \mathcal{S}$. Il est clair que cette solution satisfait $|\mathcal{S}| = r$, $\bigcup_{b \in \mathcal{S}} \mathcal{U}_b = \mathcal{U}$, et $\mathcal{U}_b \cap \mathcal{U}_{b'} = \emptyset$ pour tout $b \neq b'$. Il reste à montrer que $R_{b \rightarrow u}(\mathcal{S}) \geq R^{\min} k_b$.

Pour $b \in \mathcal{S}$ et $u \in \mathcal{U}_b$, nous avons

$$R_{b \rightarrow u}(\mathcal{S}) = \log_2 \left(1 + \frac{g_{b \rightarrow u}}{1 + \sum_{\substack{b' \in \mathcal{S} \\ b' \neq b}} g_{b' \rightarrow u}} \right) \quad (3.8)$$

$$= \log_2 \left(1 + \frac{7k/3}{1 + (k/3 - 1)} \right) \quad (3.9)$$

$$= \log_2 (1 + 7) \quad (3.10)$$

$$= 3 \quad (3.11)$$

$$\geq R^{\min} k_b, \quad (3.12)$$

où (3.9) est vraie car pour tout $b' \in \mathcal{S} \setminus \{b\}$, $g_{b' \rightarrow u} = 1$ et il existe $k/3 - 1$ termes dans

la sommation (3.8). Aussi, (3.12) est vraie car $k_b = c_b = 3$ et $R^{min} = 1$ bits/s/Hz.

- Pour le sens de *si*, supposons que FBOAH est résolu. Autrement dit, il existe un sous-ensemble $\mathcal{S}$ de β stations de base et une collection $\{\mathcal{U}_b, b \in \mathcal{S}\}$ d'ensembles d'utilisateurs tels que $\mathcal{U}_b \cap \mathcal{U}_{b'} = \emptyset$ pour $b \neq b'$, $\bigcup_{b \in \mathcal{S}} \mathcal{U}_b = \mathcal{U}$ et $R_{b \rightarrow u}(\mathcal{S}) \geq R^{min} k_b$ pour tout $b \in \mathcal{S}$ et $u \in \mathcal{U}_b$. Une solution de 3EC consiste à choisir $\mathcal{C}_b = \mathcal{U}_b$ pour tout $b \in \mathcal{S}$. Puisque $\mathcal{U}_b$ est une solution de FBOAH, alors nous devons avoir $\bigcup_{b \in \mathcal{S}} \mathcal{C}_b = \mathcal{X}$. Il reste à montrer que chaque utilisateur u est couvert par un seul ensemble, c.-à-d., $\mathcal{C}_b \cap \mathcal{C}_{b'} = \emptyset$ pour $b \neq b'$. Un utilisateur couvert u ne peut pas appartenir aux deux ensembles $\mathcal{U}_b$ et $\mathcal{U}_{b'}$ parce que, sinon, nous obtenons une contradiction. Supposons que c'est le cas, alors

$$R_{b \rightarrow u}(\mathcal{S}) = \log_2 \left(1 + \frac{g_{b \rightarrow u}}{1 + \sum_{\substack{b' \in \mathcal{S} \\ b' \neq b}} g_{b' \rightarrow u}} \right) \quad (3.13)$$

$$\leq \log_2 \left(1 + \frac{7k/3}{1 + 7k/3} \right) \quad (3.14)$$

$$< 1 \quad (3.15)$$

$$< R^{min} k_b \quad (3.16)$$

où (3.14) est vraie parce que l'utilisateur u est couvert par deux stations de base b et b' et donc la somme dans le dénominateur de (3.13) contient un terme égal à $\frac{7}{3}k$.

Ceci contredit le fait que $\mathcal{S}$ et $\mathcal{U}_b$ représentent une solution faisable de FBOAH.

En résumé, 3EC se réduit à FBOAH en temps polynomial et puisque 3EC est NP-complet, alors FBOAH est NP-difficile. Étant donnée une solution de FBOAH, nous pouvons vérifier en temps polynomial qu'elle s'agit bien d'une solution valide en vérifiant les contraintes du problème (3.7). Donc, FBOAH est NP-complet. Ceci complète la démonstration du théorème. ■

Puisque BOAH est la version d'optimisation de FBOAH, donc BOAH doit être aussi difficile que ce dernier. Donc, une conséquence directe du théorème 3.5.3 est que BOAH est NP-difficile.

3.5.2 Solutions proposées

Afin de résoudre BOAH, nous proposons de le réduire à un problème plus simple. Plus spécifiquement, nous réduisons BOAH à un problème d'ON/OFF des stations de base seulement. En d'autres termes, après avoir trouvé l'ensemble des stations de base ON, le problème d'association des utilisateurs devient un problème polynomial (c.-à-d., appartient à la classe de complexité P) comme le montre le théorème suivant.

Théorème 3.5.4 *Si l'ensemble des stations de base ON est connu, alors trouver une solution optimale de BOAH peut être fait en temps polynomial.*

Démonstration : supposons que $\mathcal{S}$ désigne l'ensemble des stations de base ON. Prenons un utilisateur arbitraire $u \in \mathcal{U}$. Premièrement, nous montrons qu'il existe au plus une station de base $b^* \in \mathcal{S}$ auquel l'utilisateur u peut être associé. Ensuite, nous montrons que trouver une telle station de base $b^* \in \mathcal{S}$ (si elle existe) peut être fait en temps polynomial. Nous procédons par contradiction. Supposons qu'il y a deux stations de base $b' \in \mathcal{S}$ et $b'' \in \mathcal{S}$ auxquelles l'utilisateur u peut être associé. Cela signifie que les débits de transmission entre u et b' et entre u et b'' sont supérieurs à R^{min} et donc supérieurs à 1 (puisque R^{min} est supérieur à 1 bits/s/Hz). Ainsi, les rapports SINR entre l'utilisateur u et les stations de base b' et b'' sont supérieurs à 1. Mathématiquement, si $R_{b' \rightarrow u}(\mathcal{S}) = \log_2(1 + \text{SINR}_{b' \rightarrow u}(\mathcal{S})) \geq 1$ et $R_{b'' \rightarrow u}(\mathcal{S}) = \log_2(1 + \text{SINR}_{b'' \rightarrow u}(\mathcal{S})) \geq 1$, alors $1 + \text{SINR}_{b' \rightarrow u}(\mathcal{S}) \geq 2$ et $1 + \text{SINR}_{b'' \rightarrow u}(\mathcal{S}) \geq 2$. Cela implique que $\text{SINR}_{b' \rightarrow u}(\mathcal{S}) \geq 1$ et $\text{SINR}_{b'' \rightarrow u}(\mathcal{S}) \geq 1$. En d'autres termes, si l'utilisateur u peut être associé à la station de base b' et b'' , alors ces deux stations de base doivent avoir un SNR supérieur à la somme

des SNR de toutes les autres stations de base. Par conséquent, nous avons

$$\text{SINR}_{b' \rightarrow u}(\mathcal{S}) = \frac{g_{b' \rightarrow u}}{1 + \sum_{\substack{j \in \mathcal{S} \\ j \neq b'}} g_{j \rightarrow u}} \geq 1, \quad (3.17)$$

et

$$\text{SINR}_{b'' \rightarrow u}(\mathcal{S}) = \frac{g_{b'' \rightarrow u}}{1 + \sum_{\substack{j \in \mathcal{S} \\ j \neq b''}} g_{j \rightarrow u}} \geq 1 \quad (3.18)$$

En développant (3.17), nous obtenons

$$1 + g_{b'' \rightarrow u} + \sum_{\substack{j \in \mathcal{S} \\ j \neq b', b''}} g_{j \rightarrow u} \leq g_{b' \rightarrow u} \quad (3.19)$$

$$\Rightarrow \frac{g_{b'' \rightarrow u} g_{b' \rightarrow u}}{1 + \sum_{\substack{j \in \mathcal{S} \\ j \neq b''}} g_{j \rightarrow u}} < 1 + g_{b'' \rightarrow u} + \sum_{\substack{j \in \mathcal{S} \\ j \neq b', b''}} g_{j \rightarrow u} \leq g_{b' \rightarrow u} \quad (3.20)$$

$$\Rightarrow \frac{g_{b'' \rightarrow u}}{1 + \sum_{\substack{j \in \mathcal{S} \\ j \neq b''}} g_{j \rightarrow u}} < 1, \quad (3.21)$$

où (3.20) est due au fait que $\frac{XY}{1+Y+Z} < 1+X+Z$ pour $X, Y, Z > 0$ avec $X = g_{b'' \rightarrow u}$, $Y = g_{b' \rightarrow u}$ et $Z = \sum_{j \in \mathcal{S}, j \neq b', b''} g_{j \rightarrow u}$.

Nous pouvons voir que (3.18) contredit (3.21). Donc, l'utilisateur u ne peut pas être associé à la station de base b' et b'' . Enfin, nous concluons que, étant donné l'ensemble $\mathcal{S}$ des stations de base ON, l'utilisateur u peut soit être associé à une station de base b^* afin de satisfaire son débit minimum R^{min} ou ne peut pas être associé du tout. Par conséquent, étant donné l'ensemble des stations de base $\mathcal{S}$, l'association des utilisateurs peut être obtenue comme suit :

- (i) calculer les débits $R_{b \rightarrow u}(\mathcal{S})$ pour tout $(b, u) \in \mathcal{B} \times \mathcal{U}$;
- (ii) pour chaque utilisateur $u \in \mathcal{U}$, trouver la station de base b (le cas échéant) tel que $R_{b \rightarrow u}(\mathcal{S}) \geq 1$; et

(iii) fixer $x_{b \rightarrow u} = 1$.

La complexité algorithmique asymptotique de cette procédure est $\mathcal{O}(kn^2)$, qui est polynomiale. Cela prouve le théorème 3.5.4. ■

Remarque 3.5.5 *BOAH donné dans (3.7) peut être reformulé comme un programme linéaire en nombres entiers comme suit. Le programme linéaire en nombres entiers ci-dessous trouve un ensemble $\mathcal{S}$ de stations de base ON de cardinalité minimale de telle sorte que $R_{b \rightarrow u}(\mathcal{S}) \geq R^{min}$ pour tout $b, u \in \mathcal{B} \times \mathcal{U}$.*

$$\text{minimiser} \quad \sum_{b=1}^n y_b \quad (3.22a)$$

$$\text{sous les contraintes} \quad x_{b \rightarrow u}, y_b \in \{0, 1\}, \forall b \in \mathcal{B}, u \in \mathcal{U}, \quad (3.22b)$$

$$\sum_{b \in \mathcal{B}} x_{b \rightarrow u} = 1, \forall u \in \mathcal{U}, \quad (3.22c)$$

$$x_{b \rightarrow u} \leq y_b, \forall b \in \mathcal{B}, u \in \mathcal{U}, \quad (3.22d)$$

$$\sum_{b' \neq b} g_{b' \rightarrow u} y_{b'} \leq x_{b \rightarrow u} \left(\frac{g_{b \rightarrow u}}{2^{R^{min}} - 1} - 1 \right) + M(1 - x_{b \rightarrow u}),$$

$$\forall b \in \mathcal{B}, u \in \mathcal{U}. \quad (3.22e)$$

où $M = \max_{b,u} (\sum_{b' \neq b} g_{b' \rightarrow u})$ est une constante utilisée pour linéariser (3.22). En résolvant (3.22) et en générant toutes les solutions possibles, il est clair que nous pouvons trouver une solution optimale de BOAH.

Dans ce qui suit, d'abord, nous discutons la procédure que nous avons développée pour vérifier la faisabilité de BOAH, puis nous décrivons les algorithmes proposés pour résoudre BOAH. Soit $\mathcal{U}_b$ l'ensemble des utilisateurs associés à la station de base b , c'est-à-dire, $\mathcal{U}_b = \{u \in \mathcal{U} : R_{b \rightarrow u}(\mathcal{S}) \geq R^{min}\}$, où $\mathcal{S}$ est l'ensemble des stations de base ON. En se

basant sur le théorème 3.5.4, l'ensemble $\mathcal{U}_b$ peut être facilement calculé pour tout $b \in \mathcal{S}$. Par conséquent, nous pouvons décider si BOAH est faisable ou non comme suit

- (a) S'il existe un utilisateur u qui n'est présent dans aucun ensemble $\mathcal{U}_b$ (c'est-à-dire, l'union des ensembles $\mathcal{U}_b$ n'est pas égale à $\mathcal{U}$), alors BOAH est non-faisable.
- (b) Sinon, s'il existe un utilisateur $u \in \mathcal{U}_b$ tel que $R_{b \rightarrow u}(\mathcal{S}) < R^{\min} k_b$, alors BOAH est non-faisable.
- (c) Sinon, BOAH est faisable.

La procédure ci-dessus nécessite la connaissance de l'ensemble $\mathcal{S}$ des stations de base ON. Cependant, trouver cet ensemble est loin d'être une tâche facile vue la NP-difficulté de BOAH. Dans ce qui suit, nous présentons un algorithme optimal et un algorithme heuristique pour trouver cet ensemble.

3.5.2.1 Algorithme optimal de force brute

Un algorithme simple qui résout BOAH consiste à énumérer toutes les combinaisons possibles d'ON/OFF des stations de base et d'association des utilisateurs. Pour chaque combinaison, cet algorithme vérifie les contraintes du problème (3.7) et détermine si cette combinaison est une solution faisable ou non. Cet algorithme nécessiterait au moins $\mathcal{O}(kn^{k+1})$ itérations dans le pire cas où $\mathcal{O}(kn)$ est la complexité de vérification des contraintes et $\mathcal{O}(n^k)$ est la complexité d'énumération de toutes les combinaisons.

Heureusement, selon le théorème 3.5.4, la complexité de cet algorithme de force brute peut être réduite. Un algorithme de force brute amélioré (appelé ci-après OPT-BF) n'énumère que les combinaisons des sous-ensembles des stations de base (c.-à-d. il n'énumère pas les combinaisons d'association des utilisateurs). Après avoir généré tous les sous-ensembles possibles des stations de base, OPT-BF vérifie les contraintes de chaque sous-ensemble et trouve l'association correspondante des utilisateurs. Par conséquent, la complexité de OPT-BF est $\mathcal{O}(kn2^n)$. Nous pouvons réduire davantage cette complexité

comme suit. Puisque dans toute solution faisable, il devrait y avoir au plus $m \triangleq \min(k, n)$ stations de base ON, la complexité de OPT-BF sera égale à

$$O\left(kn \sum_{i=1}^m \binom{n}{i}\right). \quad (3.23)$$

La complexité de OPT-BF est toujours exponentielle comme le montre (3.23). Par conséquent, ni OPT-BF ni un algorithme de séparation et d'évaluation qui résout le programme mathématique (3.7) ne peuvent efficacement trouver une solution à BOAH en temps polynomial. Pour cette raison, nous développons des algorithmes heuristiques polynomiaux. Tout d'abord, nous commençons par un simple algorithme de référence utilisé pour des raisons de comparaison. Ensuite, nous présentons l'algorithme proposé.

3.5.2.2 Un algorithme de référence simple

L'algorithme de référence est celui qui associe les utilisateurs à la station de base la plus proche, nommé ALG-CB. Autrement dit, ALG-CB associe l'utilisateur u à la station de base ayant le plus grand gain de canal. Donc, l'ensemble des stations de base ON peut s'écrire comme

$$\mathcal{S}_1 = \left\{ b \in \mathcal{B} : b = \arg \max_{b' \in \mathcal{B}} \{g_{b' \rightarrow u}\}, u \in \mathcal{U} \right\}, \quad (3.24)$$

où $|\mathcal{S}_1| \triangleq n_1$.

3.5.2.3 Un algorithme heuristique

L'algorithme proposé pour résoudre BOAH (appelé ALG-BOAH) fonctionne de manière itérative où à chaque itération il ajoute une nouvelle station de base à l'ensemble des stations de base ON. Le pseudo-code de cet algorithme est donné dans l'algorithme 5.

Algorithme 5 L'algorithme ALG-BOAH

```

1: Trouver  $\mathcal{S}_1$  en utilisant ALG-CB
2: Soit  $(1, 2, \dots, n_1)$  l'ensemble des stations de base ON triées selon (3.25)
3:  $\mathcal{S}_2 \leftarrow \emptyset$  //  $|\mathcal{S}_2| = n_2$ 
4: pour  $i \leftarrow 1$  à  $n_1$  faire
5:    $\mathcal{S}_2 \leftarrow \{i\}$ 
6:   résultat  $\leftarrow$  vérifierdébit( $\mathcal{S}_2$ )
7:   tant que résultat = 0 et  $i \leq n_2 - 1$  faire
8:     pour  $j \leftarrow i + 1$  à  $n_1$  faire
9:        $\mathcal{S}_2 \leftarrow \mathcal{S}_2 \cup \{j\}$ 
10:      résultat  $\leftarrow$  vérifierdébit( $\mathcal{S}_2$ )
11:      si résultat  $\neq 0$  alors
12:        retourner  $\mathcal{S}_2$ 
13:      sinon
14:         $\mathcal{S}_2 \leftarrow \mathcal{S}_2 \setminus \{j\}$ 
15:      fin si
16:    fin pour
17:     $i \leftarrow i + 1$ 
18:     $\mathcal{S}_2 \leftarrow \mathcal{S}_2 \cup \{i\}$ 
19:    résultat  $\leftarrow$  vérifierdébit( $\mathcal{S}_2$ )
20:  fin tant que
21: fin pour
22: retourner  $\emptyset$ 

```

ALG-BOAH fait appel à une procédure appelée « vérifierdébit($\mathcal{S}_2$) » qui accepte comme entrée un ensemble $\mathcal{S}_2$ (de cardinalité n_2 , c.-à-d., $|\mathcal{S}_2| = n_2$) de stations de base ON et vérifie les exigences des débits des utilisateurs. Si tous les utilisateurs sont satisfaits, elle retourne 1 et sinon elle retourne 0. ALG-BOAH commence par trouver l'ensemble de stations de base $\mathcal{S}_1$ retourné par ALG-CB. Ensuite, il trie cet ensemble en fonction de la somme des gains des canaux. L'ensemble trié est donné par n_1 -uplet $(1, 2, \dots, n_1)$, où

$$\sum_{u=1}^k g_{1 \rightarrow u} \geq \sum_{u=1}^k g_{2 \rightarrow u} \geq \dots \geq \sum_{u=1}^k g_{n_1 \rightarrow u}. \quad (3.25)$$

Une fois le tri terminé, ALG-BOAH commence à activer la première station de base. Si tous les utilisateurs peuvent y être associés, l'algorithme se termine. Sinon, ALG-BOAH

crée l'ensemble $\mathcal{S}_2 = \{1, j\}$ de deux stations de base et vérifie les débits minima des utilisateurs pour chaque j dans $\{2, \dots, n_1\}$. Si cela échoue, il crée l'ensemble $\mathcal{S}_2 = \{1, 2, j\}$ de trois stations de base et vérifie les débits minima des utilisateurs pour chaque j dans $\{3, \dots, n_1\}$. Cette procédure se poursuit de la même manière jusqu'à ce que $\mathcal{S}_2 = \{1, 2, \dots, n_1\}$. Toutes ces combinaisons sont choisies de manière à conserver la meilleure station de base avant de passer aux itérations suivantes. Dans le cas où ALG-BOAH retourne l'ensemble vide $\mathcal{S}_2 = \emptyset$, il conclut que BOAH est non-faisable.

Le temps d'exécution de ALG-BOAH est clairement polynomial. En fait, sa complexité dans le pire cas est donnée par $\mathcal{O}(knf(n_1))$ où $f(n_1)$ est une fonction polynomiale de n_1 qui représente la complexité de génération des combinaisons et $\mathcal{O}(kn)$ est la complexité de la procédure « vérifierdébit($\mathcal{S}_2$) ». Explicitement, la fonction $f(n_1)$ est donnée par :

$$f(n_1) = \sum_{i=1}^{n_1-1} \sum_{j=1}^{n_1-i} j = \frac{n_1(n_1-1)(2n_1-1)}{12} + \frac{n_1(n_1-1)}{4} = \mathcal{O}(n_1^3).$$

Par conséquent, la complexité globale dans le pire cas de ALG-BOAH est $\mathcal{O}(knn_1^3)$. Puisque $n_1 \leq m$, la complexité globale de ALG-BOAH est

$$\mathcal{O}(knm^3). \quad (3.26)$$

3.6 BOA avec débits minima faibles

Cette section étudie BOA dans le cas où $R^{min} < 1$ bits/s/Hz (appelé BOAL). Dans ce cas, BOA devient très difficile puisque le problème ON/OFF des stations de base et d'association des utilisateurs ne peuvent pas être découplés, c'est-à-dire que le théorème 3.5.4 ne peut pas être appliqué. Afin de résoudre efficacement BOAL, d'abord nous utilisons une relaxation largement utilisée dans la littérature (Pollakis *et al.*, 2012; Zhuang *et al.*, 2016). Cette relaxation utilise le pire scénario d'interférence en supposant que toutes les

stations de base interfèrent entre-elles. Le débit de transmission entre la station de base b et l'utilisateur u devient :

$$R_{b \rightarrow u} = \log_2 \left(1 + \frac{p_b^r h_{b \rightarrow u}}{\sigma^2 + \sum_{\substack{b' \in \mathcal{B} \\ b' \neq b}} p_{b'}^r h_{b' \rightarrow u}} \right), \text{ [en bits/s/Hz]} \quad (3.27)$$

En d'autres termes, le débit de transmission $R_{b \rightarrow u}$ défini dans (3.27) peut être considéré comme une limite inférieure sur le débit de transmission réel $R_{b \rightarrow u}(\mathbf{y})$ défini dans (3.5). Avec cette relaxation, BOAL peut être reformulé comme suit :

$$\text{minimiser} \quad \sum_{b=1}^n y_b \quad (3.28a)$$

$$\text{sous les contraintes} \quad x_{b \rightarrow u}, y_b \in \{0, 1\}, \forall b \in \mathcal{B}, u \in \mathcal{U}, \quad (3.28b)$$

$$x_{b \rightarrow u} \leq y_b, \forall b \in \mathcal{B}, u \in \mathcal{U}, \quad (3.28c)$$

$$\sum_{b \in \mathcal{B}} x_{b \rightarrow u} = 1, \forall u \in \mathcal{U}, \quad (3.28d)$$

$$\sum_{u' \in \mathcal{U}} x_{b \rightarrow u'} \leq \bar{R}_{b \rightarrow u} x_{b \rightarrow u} + k(1 - x_{b \rightarrow u}), \forall b \in \mathcal{B}, u \in \mathcal{U}. \quad (3.28e)$$

où $\bar{R}_{b \rightarrow u}$ est défini comme le débit de transmission *normalisé*, soit

$$\bar{R}_{b \rightarrow u} \triangleq \frac{R_{b \rightarrow u}}{R^{min}},$$

et la valeur k est introduite dans les contraintes (3.28e) afin de linéariser le problème. Bien que (3.28) peut être efficacement résolu en utilisant des solveurs comme (IBM, 2010; Gurobi Optimization Inc., 2016), théoriquement, les algorithmes mis en oeuvre dans des tels solveurs requièrent une complexité exponentielle dans le pire cas. Pour résoudre efficacement BOAL, nous commençons par étudier sa complexité computationnelle, puis nous présentons nos algorithmes proposés.

3.6.1 NP-difficulté

Dans cette partie, nous montrons que trouver une solution faisable à BOAL est NP-complet. La démonstration est basée sur une réduction polynomiale du problème de couverture par ensembles (appelé SC) à la version de faisabilité de BOAL (appelée FBOAL). Le problème de couverture par ensembles est défini ci-après.

Définition 3.6.1 (Couverture par ensembles (SC) (Garey et Johnson, 1979)) :

INSTANCE : un ensemble $\mathcal{X}$ de x entiers, appelé l'univers, un entier d , et une collection $\mathcal{C} \triangleq \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_c\}$ de $c \geq d$ sous-ensembles de $\mathcal{X}$ dont l'union est égale à l'univers.

QUESTION : existe-t-il une sous-collection $\mathcal{C}^*$ de d sous-ensembles de $\mathcal{C}$ qui couvre $\mathcal{X}$? C'est-à-dire, l'union des sous-ensembles de $\mathcal{C}^*$ est $\mathcal{X}$.

Il est bien connu que SC est NP-complet (Garey et Johnson, 1979). Le problème FBOAL est défini comme

Définition 3.6.2 (FBOAL) :

INSTANCE : un ensemble $\mathcal{B}$ de stations de base, un ensemble $\mathcal{U}$ d'utilisateurs, un entier positif $\beta \leq n$, et une matrice de débits de transmission normalisés $\bar{\mathbf{R}} = [\bar{R}_{b \rightarrow u}]$.

QUESTION : existe-t-il un sous-ensemble $\mathcal{S}$ de $\mathcal{B}$ de β stations de base et une collection $\{\mathcal{U}_b, b \in \mathcal{S}\}$ d'ensembles d'utilisateurs tel que $\mathcal{U}_b \cap \mathcal{U}_{b'} = \emptyset$ pour $b \neq b'$, $\bigcup_{b \in \mathcal{S}} \mathcal{U}_b = \mathcal{U}$ et $\bar{R}_{b \rightarrow u} \geq k_b$ pour tout $b \in \mathcal{S}$ et $u \in \mathcal{U}_b$ (ou $k_b \triangleq |\mathcal{U}_b|$)?

Contrairement à BOAH, nous choisissons ici SC (au lieu de 3EC) car nous supposons le pire scénario d'interférence et, comme nous verrons bientôt, nous n'avons pas besoin de restreindre le nombre d'utilisateurs par station de base à 3.

Théorème 3.6.3 *FBOAL est NP-complet.*

Démonstration : soit $I = (\mathcal{X}, \mathcal{C})$ une instance de SC. L'instance I' de BOAL est construite comme suit

1. Soit $\mathcal{U} = \mathcal{X}$, $n = c$ et $\beta = d$.
2. Soit $\bar{R}_{b \rightarrow u}$ la matrice suivante :

$$\bar{R}_{b \rightarrow u} \triangleq \begin{cases} 2k, & \text{si } u \in \mathcal{C}_b \\ 1, & \text{sinon} \end{cases}$$

La valeur $2k$ est choisie arbitrairement pour garantir qu'un utilisateur n'est pas associé à plusieurs stations de base et que son débit minimum est respecté.

Nous affirmons qu'il existe une couverture de taille d si et seulement si au plus β stations de base peuvent satisfaire les k utilisateurs, donc résolvent FBOAL.

- Pour le sens de *seulement si*, supposons qu'il existe une couverture par ensembles $\{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ de taille d . Soient $\mathcal{S} = \{1, \dots, d\}$ et $\mathcal{U}_b = \mathcal{C}_b$ pour tout $b \in \mathcal{S}$. Les deux ensembles $\mathcal{S}$ et $\mathcal{U}_b$ constituent une solution à FBOAL. En effet, par construction, nous avons $\bigcup_{b \in \mathcal{S}} \mathcal{U}_b = \mathcal{U}$ et, si $u \in \mathcal{U}_b$ pour une certaine b , alors nous avons $\bar{R}_{b \rightarrow u} = 2k \geq k \geq k_b$. Cependant, cette solution de FBOAL peut couvrir un utilisateur plus d'une fois; ce qui ne respecte pas les contraintes de BOAL. Si c'est le cas, nous pouvons chercher chaque utilisateur associé à plusieurs stations de base et nous l'associons à une seule station de base arbitrairement. Puisque cette procédure ne peut pas augmenter la taille de $\mathcal{S}$, donc nous concluons qu'il existe au plus d stations de base qui résolvent FBOAL.
- Pour le sens de *si*, supposons qu'il y ait au plus β stations de base qui résolvent FBOAL. C'est-à-dire, il existe un ensemble $\mathcal{S} = \{1, \dots, \beta\}$ de stations de base et des ensembles disjoints $\{\mathcal{U}_b, b \in \mathcal{S}\}$ satisfaisant $\bigcup_{b \in \mathcal{S}} \mathcal{U}_b = \mathcal{U}$ et $\bar{R}_{b \rightarrow u} \geq k_b$ pour tout $b \in \mathcal{S}$ et $u \in \mathcal{U}_b$. Soit $\mathcal{C}^* = \{\mathcal{U}_1, \dots, \mathcal{U}_\beta\}$. Cette collection constitue une solution à SC car elle est une sous-collection de $\beta = d$ ensembles qui couvre $\mathcal{C}$. Par conséquent,

SC est résolu.

Puisque la réduction est faite en temps polynomial et que SC est NP-complet, alors FBOAL est NP-difficile. Nous pouvons vérifier en temps polynomial qu'une solution à FBOAL est valide. Donc, FBOAL est dans NP et par la suite FBOAL est NP-complet. Ceci termine la démonstration du théorème 3.6.3. ■

Le théorème 3.6.3 prouve qu'il est NP-complet de décider s'il existe ou non une solution faisable à BOAL. Évidemment, trouver une meilleure solution serait plus difficile. Par conséquent, BOAL est NP-difficile.

En raison de la NP-difficulté de BOAL, nous proposons un algorithme glouton simple. Puisque cet algorithme ne peut pas toujours trouver une solution faisable (il ne peut pas toujours associer tous les utilisateurs et garantir leurs débits minima), nous proposons de reformuler BOAL comme un problème de contrôle d'admission afin de sélectionner les meilleurs utilisateurs. Nous montrons que l'algorithme proposé a deux propriétés : (i) quand il trouve une solution faisable, sa valeur (le nombre de stations de base ON) est majoré par $\mathcal{O}(\log k)$ fois le nombre optimal, et (ii) lorsqu'il ne trouve pas une solution faisable, sa valeur (le nombre d'utilisateurs associés) est minoré par $\mathcal{O}(1)$ fois le nombre optimal.

3.6.2 Solutions proposées

Dans cette partie, nous supposons que BOAL est faisable. L'idée principale de l'algorithme proposé pour résoudre BOAL (appelé ALG-BOAL) est de commencer par la station de base qui peut associer autant d'utilisateurs que possible. Ceci permet de maximiser le nombre d'utilisateurs associés à chaque station de base et donc permet de minimiser le nombre de stations de base ON. Le pseudo-code de ALG-BOAL est donné dans l'algorithme 6. Nous utilisons $\bar{\mathbf{R}}_u$ (resp. $\bar{\mathbf{R}}^b$) pour noter la u -ème ligne (resp. la b -ème colonne)

de la matrice $\bar{\mathbf{R}} = [\bar{R}_{b \rightarrow u}]$ des débits de transmission normalisés.

Dans ALG-BOAL, à chaque station de base b est assignée un poids w_b . Ce poids est égal au plus grand nombre d'utilisateurs qui ont un débit normalisé supérieur ou égal à w_b . Par exemple, s'il y a quatre utilisateurs et le débit normalisé de b est donné par $[1, 2, 3, 4]$, alors $w_b = 2$; il y a seulement **deux** utilisateurs qui ont un débit normalisé supérieur à 2 (il n'y a pas **trois** utilisateurs qui ont un débit normalisé supérieur à 3). Autrement dit, w_b est le plus grand entier tel qu'il existe un ensemble $\mathcal{W}$ de w_b utilisateurs et $w_b \leq \bar{R}_{b \rightarrow u}$ pour tout $u \in \mathcal{W}$. Après avoir trié la b -ème colonne $\bar{\mathbf{R}}^b$ par ordre décroissant, le poids de b peut être calculé comme suit :

$$w_b = \max_{u \in \mathcal{U}} (\min(\bar{R}_{b \rightarrow u}, u)). \quad (3.29)$$

Nous utilisons la notation vectorielle suivante $\mathbf{w} \triangleq [w_1, w_2, \dots, w_n]$.

ALG-BOAL commence par trouver la station de base b^* qui a le poids maximal. Ensuite, il itère les stations de base et associe le plus grand nombre d'utilisateurs à chaque station de base tant que leurs débits minima sont respectés. À chaque itération, ALG-BOAL supprime de l'ensemble des utilisateurs ceux déjà associés et met à jour les poids des stations de base. La station de base b^* associe les utilisateurs en fonction de la valeur de leur débits de transmission normalisés $\bar{R}_{b \rightarrow u}$ en commençant par celui qui a la plus grande valeur. ALG-BOAL continue de cette manière jusqu'à ce que tous les utilisateurs soient associés ou que toutes les stations de base soient déjà ON ou qu'aucune solution ne soit trouvée.

Remarque 3.6.4 *Au niveau de la ligne 5 de l'algorithme 6, ALG-BOAL utilise une règle du « i -ème minimum » en cas d'égalité, c.-à-d. lorsqu'il existe plusieurs stations de base qui ont la même valeur maximale de poids, ALG-BOAL choisit celle qui a le i -ème plus petit élément dans $\bar{\mathbf{R}}^b$. En d'autres termes, si $w_b = w_{b'}$, alors ALG-BOAL choisit b avant b' dans le tri, si le i -ème plus petit élément se trouve dans $\bar{\mathbf{R}}^b$, et il choisit b' avant b sinon. Par exemple, supposons que $\bar{\mathbf{R}}^1 = (7, 4, 1)$ et $\bar{\mathbf{R}}^2 = (3, 5, 1)$. Nous avons un cas d'égalité déjà puisque*

Algorithme 6 L'algorithme ALG-BOAL

```

1: Calculer les poids selon (3.29)
2:  $u \leftarrow 1$ 
3:  $\mathcal{S} \leftarrow \emptyset$ 
4: tant que  $u \leq k$  and  $|\mathcal{S}| \leq n$  faire
5:    $b^* \leftarrow \arg \max(\mathbf{w})$ 
6:    $\mathcal{U}_{b^*} \leftarrow \emptyset$  //  $k^* \triangleq |\mathcal{U}_{b^*}|$ 
7:    $u^* \leftarrow \arg \max(\bar{\mathbf{R}}^{b^*})$ 
8:   tant que  $\bar{R}_{b^* \rightarrow u^*} \geq k_{b^*} + 1$  faire
9:      $\mathcal{U}_{b^*} \leftarrow \mathcal{U}_{b^*} \cup \{u^*\}$ 
10:     $\bar{\mathbf{R}}_{u^*} \leftarrow 0$ 
11:     $u^* \leftarrow \arg \max(\bar{\mathbf{R}}^{b^*})$ 
12:   fin tant que
13:   si  $\mathcal{U}_{b^*} \neq \emptyset$  alors
14:      $\mathcal{S} \leftarrow \mathcal{S} \cup \{b^*\}$ 
15:   fin si
16:   Mettre à jour les poids selon (3.29)
17:    $u \leftarrow u + 1$ 
18: fin tant que
19: retourner  $\mathcal{S}$  et  $\mathcal{X} \triangleq \{\mathcal{U}_b, b \in \mathcal{S}\}$ 

```

$w_1 = w_2 = 2$. Le premier minimum de $\bar{\mathbf{R}}^1$ (de la station de base 1) est égal au premier minimum de $\bar{\mathbf{R}}^2$ (de la station de base 2). ALG-BOAL vérifie ainsi le deuxième minimum qui est égal à 3 dans $\bar{\mathbf{R}}^2$ mais égal à 4 dans $\bar{\mathbf{R}}^1$. Donc, ALG-BOAL choisit la station de base 2 avant 1 dans l'ordre de tri.

Dans ce qui suit, nous analysons les caractéristiques de ALG-BOAL.

Théorème 3.6.5 Si ALG-BOAL associe tous les utilisateurs, alors il active au plus $\mathcal{O}(\log k)o$ stations de base, où o est le nombre des stations de base on dans une solution optimale. Autrement dit, ALG-BOAL est un algorithme $\mathcal{O}(\log k)$ -approximation.

Démonstration : d'abord, nous définissons le j -ème nombre harmonique $H_j \triangleq \sum_{i=1}^j 1/i$ où $H_0 \triangleq 0$. Maintenant, supposons que ALG-BOAL trouve une solution faisable à BOAL. Soit t une itération arbitraire de ALG-BOAL et soit $\mathcal{U}_t$ l'ensemble des utilisateurs non en-

core associés au début de la t -ème itération. (Le nombre $x_t \triangleq |\mathcal{U}_t|$ représente le nombre d'utilisateurs non encore associés au début de la t -ème itération.) Si ALG-BOAL se termine après T itérations, alors nous obtenons $x_1 = k$ et $x_{T+1} = 0$. En outre, il est facile de voir que $x_t \geq x_{t+1}$ et qu'au plus o stations de base sont nécessaires pour associer x_t utilisateurs (par l'existence de o). Puisqu'à l'itération $t + 1$, ALG-BOAL associe le nombre maximum d'utilisateurs parmi $\mathcal{U}_t$, alors il doit associer au moins x_t/o utilisateurs ; s'il associe moins que ça, il n'y a aucun moyen d'associer tous les x_t utilisateurs avec o stations de base, ce qui contredit l'existence de o . Par conséquent, nous avons

$$x_t - x_{t+1} \geq \frac{x_t}{o}, \quad \forall t, \quad (3.30)$$

En multipliant les deux côtés de l'inégalité (3.30) par o/x_t et en sommant ces inégalités pour $t = 1, \dots, T$, nous obtenons :

$$\begin{aligned} \sum_{t=1}^T 1 &\leq \sum_{t=1}^T \frac{x_t - x_{t+1}}{x_t} o \\ T &\leq \left(\sum_{t=1}^T \sum_{i=1+x_{t+1}}^{x_t} \frac{1}{x_t} \right) o \\ T &\leq \left(\sum_{t=1}^T \sum_{i=1+x_{t+1}}^{x_t} \frac{1}{i} \right) o && (\text{parce que } i \leq x_t) \\ &= \left(\sum_{t=1}^T \left(\sum_{i=1}^{x_t} \frac{1}{i} - \sum_{i=1}^{x_{t+1}} \frac{1}{i} \right) \right) o \\ &= \left(\sum_{t=1}^T (H_{x_t} - H_{x_{t+1}}) \right) o \\ &= (H_{x_1} - H_{x_{T+1}}) o && (\text{parce que la somme télescope}) \\ &= (H_{x_1} - H_0) o = H_k o && (\text{parce que } H_0 = 0) \end{aligned}$$

Puisque ALG-BOAL active une station de base à chaque itération, le nombre de stations de base ON, après sa terminaison, sera égal à T . Ainsi, le nombre de stations de base ON

retourné par ALG-BOAL est au plus égal à H_k fois α . Par conséquent, ALG-BOAL est un algorithme H_k -approximation pour BOAL. Puisqu'il est bien connu que $H_k = \mathcal{O}(\log k)$. Puisque ALG-BOAL est un algorithme polynomial comme le montre le pseudo-code de l'algorithme 6, ceci prouve le théorème 3.6.5. ■

Dans ce qui suit, nous supposons qu'ALG-BOAL peut ne pas trouver une solution faisable à BOAL. Donc, il est important d'appliquer une procédure de contrôle d'admission qui élimine le nombre minimum d'utilisateurs afin de trouver une solution faisable. Pour ce faire, nous pouvons facilement démontrer, par contradiction, que si BOAL est non-faisable avec n stations de base alors il est non-faisable avec $n - 1$ stations de base. Par conséquent, nous transformons BOAL en un problème de contrôle d'admission en maximisant le nombre d'utilisateurs associés, appelé MAX-BOAL qui est défini comme suit :

$$\text{maximiser} \quad \sum_{u=1}^k \sum_{b=1}^n x_{b \rightarrow u} \quad (3.31a)$$

$$\text{sous les contraintes} \quad x_{b \rightarrow u} \in \{0, 1\}, \forall b \in \mathcal{B}, u \in \mathcal{U}, \quad (3.31b)$$

$$\sum_{b \in \mathcal{B}} x_{b \rightarrow u} \leq 1, \forall u \in \mathcal{U}, \quad (3.31c)$$

$$\sum_{u' \in \mathcal{U}} x_{b \rightarrow u'} \leq \bar{R}_{b \rightarrow u} x_{b \rightarrow u} + k(1 - x_{b \rightarrow u}), \forall b \in \mathcal{B}, u \in \mathcal{U}. \quad (3.31d)$$

MAX-BOAL maximise le nombre d'utilisateurs associés sous les contraintes (i) des débits minima des utilisateurs tel qu'indiqué dans (3.31d) et (ii) d'au plus une station de base par utilisateur tel qu'indiqué dans (3.31c). Soit α^{max} la valeur optimale de (3.31). Dans le lemme suivant, nous illustrons la relation qui existe entre la valeur optimale de BOAL et celle de MAX-BOAL.

Théorème 3.6.6 *BOAL est faisable si et seulement si $\alpha^{max} = k$. Par conséquent, s'il existe un algorithme α -approximation ALG pour MAX-BOAL alors ALG donne une solution à*

BOAL avec au plus $(1 - \alpha)k$ utilisateurs qui ne respectent pas leurs débits minima.

Démonstration : d'une part, si BOAL est faisable, alors il existe une matrice $\mathbf{X}$ qui satisfait les contraintes (3.28d) et (3.28e). Par conséquent, selon (3.28d), tous les utilisateurs sont associés et donc $o^{max} = k$. D'autre part, si $o^{max} = k$ alors il existe une matrice $\mathbf{X}$ qui satisfait les contraintes (3.31c) et (3.31d). Le fait qu'il y ait k utilisateurs implique que les contraintes (3.31c) sont satisfaites par égalité et donc BOAL est faisable.

Supposons maintenant que ALG est un algorithme α -approximation pour MAX-BOAL. Ainsi, $A^{max} \leq o^{max} \leq \alpha A^{max}$ avec A^{max} le nombre d'utilisateurs associés par ALG. Si BOAL est faisable, alors $o^{max} = k$ et donc $A^{max} \leq k \leq \alpha A^{max}$ ou alternativement $0 \leq k - A^{max} \leq k - \alpha k$. Par conséquent, au plus $(1 - \alpha)k$ utilisateurs ne respectent pas leurs débits minima. Cela prouve le théorème. ■

Une conséquence immédiate de la NP-difficulté de BOAL et du théorème 3.6.6 est que MAX-BOAL est aussi NP-difficile. Heureusement, nous démontrons dans ce qui suit que ALG-BOAL est un algorithme 2-approximation pour MAX-BOAL.

Théorème 3.6.7 *ALG-BOAL est un algorithme 2-approximation pour MAX-BOAL.*

Démonstration : soient $\mathcal{S}$ et $\mathcal{X} = \bigcup_{b \in \mathcal{S}} \mathcal{U}_b$ l'ensemble des stations de base et l'ensemble des utilisateurs associés, respectivement, par ALG-BOAL. Soit $A^{max} \triangleq |\mathcal{X}|$. Nous supposons sans perte de généralité que $w_1 \geq w_2 \geq \dots \geq w_{|\mathcal{S}|}$. Nous démontrons que

$$o^{max} \leq 2A^{max}.$$

Dans l'algorithme optimal, aucun utilisateur i dans $\mathcal{U} \setminus \mathcal{X}$ ne peut être associé à une station de base j dans $\mathcal{B} \setminus \mathcal{S}$. En effet, si c'était le cas, alors ALG-BOAL aurait considéré une telle association. Donc, l'utilisateur i s'associe à la station de base j si (i) $i \in \mathcal{X}$ et $j \in \mathcal{B} \setminus \mathcal{S}$ ou (ii) $i \in \mathcal{U}$ et $j \in \mathcal{S}$. Dans le premier cas, l'algorithme optimal ne peut clairement pas

associer plus de A^{max} utilisateurs. Dans le second cas, supposons que l'algorithme optimal associe plus que A^{max} utilisateurs. Cela signifie qu'il existe au moins une station de base b dans $\mathcal{S}$ tel que le nombre d'utilisateurs associés à b par l'algorithme optimal est supérieur au nombre d'utilisateurs associés à b par ALG-BOAL. Comme ALG-BOAL, par définition, associe le nombre maximum d'utilisateurs possibles. Cela conduit à une contradiction, c'est-à-dire que l'algorithme optimal ne peut pas associer plus d'utilisateurs que ALG-BOAL aux stations de base de $\mathcal{S}$. Finalement, dans les deux cas, nous avons

$$O^{max} \leq 2A^{max},$$

ce qui prouve le théorème 3.6.7. ■

Finalement, ALG-BOAL peut être résumée comme suit :

- Avant le début de ALG-BOAL, nous vérifions
 - S'il existe un utilisateur u pour lequel $\bar{R}_{b \rightarrow u} < 1$ pour tout b , alors BOAL est non-faisable.
 - S'il existe un utilisateur u pour lequel il existe une unique b tel que $\bar{R}_{b \rightarrow u} \geq 1$, alors nous associons u à b , puis nous appliquons ALG-BOAL.
- Après la fin de ALG-BOAL, nous vérifions
 - Si $\bigcup_{b \in \mathcal{S}} \mathcal{U}_b \neq \mathcal{U}$, alors BOAL est non-faisable.
 - Sinon, BOAL est faisable.

Pour la complexité asymptotique de ALG-BOAL, le calcul des poids w , nécessite de trier n colonnes de $\bar{\mathbf{R}}$. Par conséquent, la ligne 1 de l'algorithme 6 requiert $\mathcal{O}(kn \log k)$ itérations dans le pire cas. La boucle **tant que** au niveau de la ligne 4 nécessite $\mathcal{O}(m)$ itérations. À partir de la ligne 5, ALG-BOAL nécessite $\mathcal{O}(k)$ ou $\mathcal{O}(n)$. Les ligne 1 et 16 sont les opérations les plus complexes avec une complexité de $\mathcal{O}(kn \log k)$. Globalement, la

complexité dans le pire cas de ALG-BOAL est

$$\mathcal{O}(knm \log k), \quad (3.32)$$

où $m \triangleq \min(k, n)$.

3.7 Résultats de simulations

Dans cette section, nous présentons les résultats de simulation pour illustrer les performances des algorithmes proposés. Nous considérons une zone géographique sous forme d'un carré de côté égal à 500 mètres où les utilisateurs et les stations de base sont distribués de manière aléatoire et uniforme. Les gains des canaux sont similaires aux spécifications 3GPP (3GPP, 2010) et modélisés de la même manière que (Zhuang *et al.*, 2016). Sauf indication contraire, $R^{\min} = 1$ bits/s/Hz pour BOAH et 0,1 bits/s/Hz pour BOAL. Tous les paramètres de simulation sont présentés dans le tableau 3.2. Les problèmes d'optimisation sont modélisés dans Python 2.7 à l'aide du modélisateur PuLP (Mitchell *et al.*, 2011) et sont résolus à l'aide du solveur CPLEX (IBM, 2010). Nous effectuons 1000 réalisations de canaux aléatoires et indépendantes et nous calculons la moyenne.

Paramètre	Valeur
Puissance de la station de base macro-cellule	46 dBm
Puissance des petites stations de base	30 dBm
Bande passante totale	10 MHz
Fréquence porteuse	2 GHz
Densité de puissance du bruit	-174 dBm/Hz
Gain du canal entre u et la macro-cellule	$15.3 + 37.6 \log_{10} d_{0 \rightarrow u}$
Gain du canal entre u et b	$30.6 + 36.7 \log_{10} d_{b \rightarrow u}$

Tableau 3.2 Paramètres de simulations pour BOA.

La faisabilité de BOA est mesurée pour chaque algorithme par le pourcentage des

réalisations non-faisable ainsi que le pourcentage des mauvaises décisions. Pour un algorithme ALG, une réalisation est considérée comme non-faisable si ALG ne trouve pas une solution, alors qu'une décision est considérée comme mauvaise si l'algorithme optimal trouve une solution faisable mais ALG n'en trouve pas. Lorsque les résultats de simulation ne concernent pas la faisabilité, nous ignorons les réalisations non-faisables et calculons la moyenne sur les réalisations faisables seulement. Pour BOAH, nous comparons ALG-BOAH à l'algorithme optimal OPT-BF et à l'algorithme de référence ALG-CB. Quant à BOAL, nous comparons ALG-BOAL à l'algorithme optimal OPT-ILP et à l'algorithme heuristique ALG-LP. OPT-ILP est obtenu en résolvant le programme linéaire en nombres entiers (3.28) en utilisant CPLEX alors que ALG-LP est un algorithme proposé dans (Pollakis *et al.*, 2012; Zhuang *et al.*, 2016) pour résoudre un problème similaire à BOAL et nous l'adaptions pour résoudre BOAL.

Nous commençons par illustrer les performances des algorithmes en termes de faisabilité. La figure 3.1 illustre les pourcentages des réalisations non-faisables ainsi que les pourcentages des mauvaises décisions prises par les algorithmes proposés pour BOAH. Il est clair que plus le nombre d'utilisateurs augmente, plus le nombre des réalisations non-faisables devient grand. Évidemment, l'algorithme optimal OPT-BF ne fait aucune mauvaise décision ; ayant ainsi un pourcentage toujours égal à 0%. Pour ALG-BOAH, le pourcentage des réalisations non-faisables et celui des mauvaises décisions sont toujours meilleurs que ceux de ALG-CB.

La figure 3.2 illustre les pourcentages des réalisations non-faisables ainsi que les pourcentages des mauvaises décisions prises par les algorithmes proposés pour BOAL. Cette figure montre une différence substantielle par rapport à la figure 3.1, car des petites valeurs de R^{min} diminuent certainement le nombre de réalisations non-faisables. ALG-BOAL trouve rarement une mauvaise décision, surtout quand k et R^{min} ne sont pas très grands. De plus, en comparant ALG-BOAL et ALG-LP, nous pouvons voir la différence énorme que

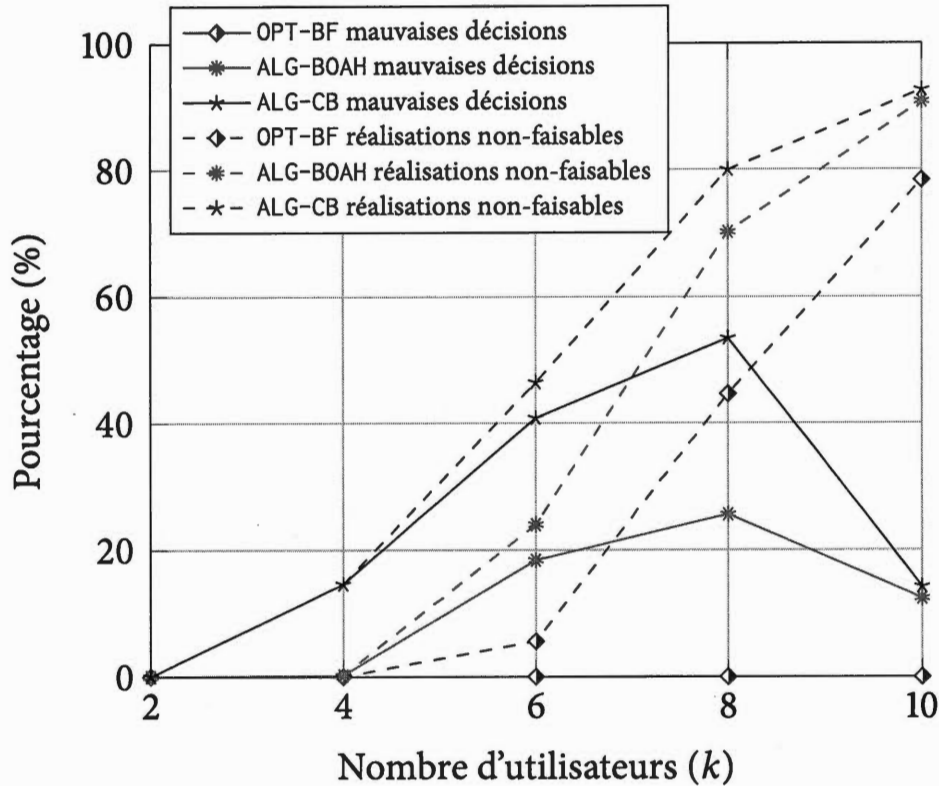


Figure 3.1 Faisabilité de BOAH en fonction de k avec $n = 16$.

ce dernier donne; près de 60% de mauvaises décisions pour $k = 30$ pour ALG-LP alors que ALG-BOAL donne seulement 10% de mauvaises décisions. Cette énorme différence est due au fait qu'ALG-LP n'est pas spécifiquement conçu pour résoudre BOAL et il n'est pas optimisé pour vérifier sa faisabilité.

Dans les figures 3.3 et 3.4, nous montrons le nombre moyen de stations de base ON retourné par les algorithmes proposés³.

La figure 3.3 illustre le nombre moyen de stations de base ON retourné par OPT-BF, ALG-BOAH et ALG-CB. Elle montre que ALG-CB consomme plus d'énergie par rapport à

3. En raison des débits minima élevés, le nombre d'utilisateurs k ne peut pas être choisi supérieur à 6 pour $n = 10$; sinon, le nombre de réalisations non-faisables sera très grand.

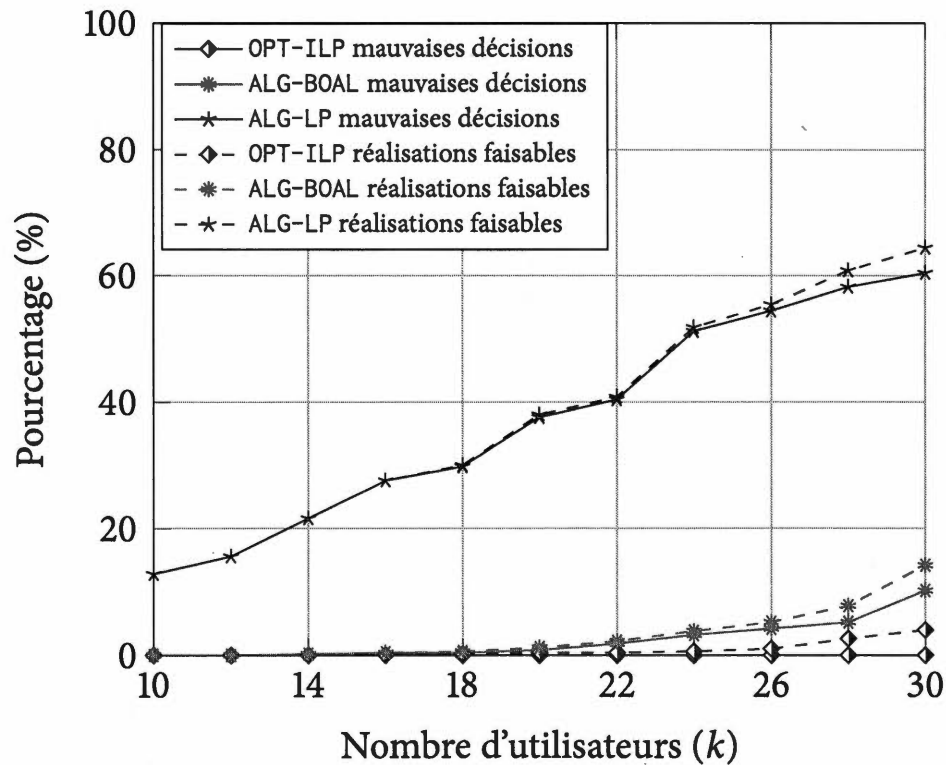


Figure 3.2 Faisabilité de BOAL en fonction de k avec $n = 16$.

ALG-BOAH car il active presque un utilisateur par station de base lorsque $n = 100$ et k est petit. Nous observons que lorsque k augmente, le nombre moyen de stations de base ON retourné par ALG-BOAH est toujours inférieur à celui retourné par ALG-CB. D'autre part, ALG-BOAH donne des performances qui s'approchent de celles d'OPT-BF lorsque $n = 10$.

La figure 3.4 illustre le nombre moyen de stations de base ON retourné par OPT-ILP, ALG-BOAL, et ALG-LP. Nous observons que ALG-BOAL atteint des performances proches de celles de l'optimal, même pour k et n élevés. De plus, ALG-BOAL atteint des meilleures performances par rapport à l'algorithme de référence ALG-LP surtout lorsque n augmente.

Maintenant, nous illustrons les durées moyennes du temps d'exécution des algo-

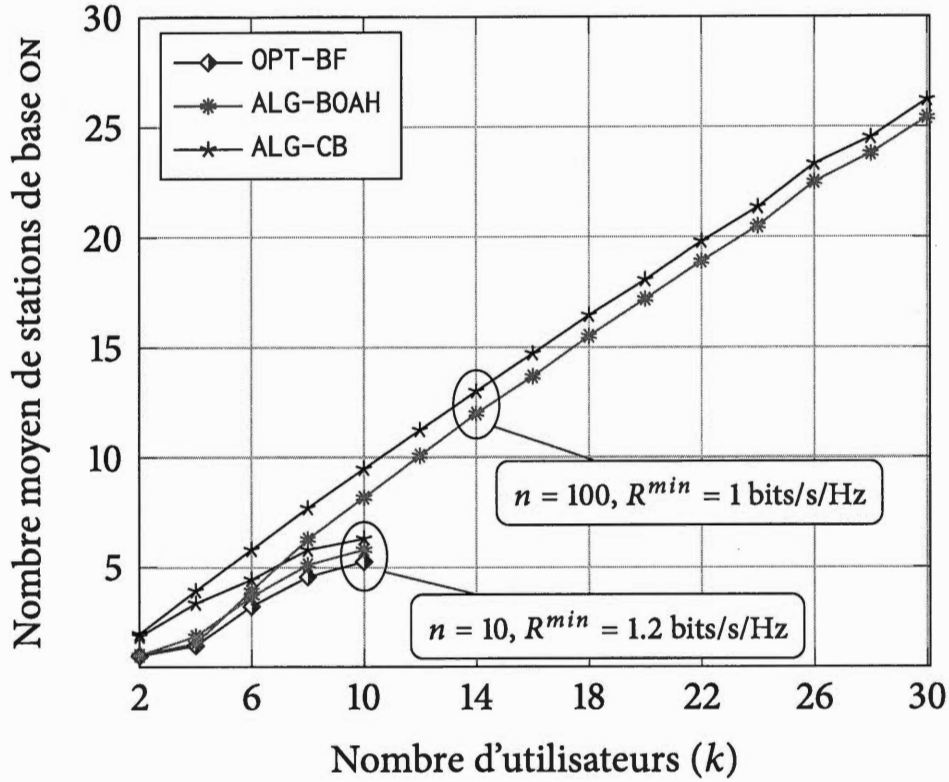


Figure 3.3 Nombre moyen de stations de base ON par tous les algorithmes en fonction de k pour BOAH.

algorithmes optimaux ainsi que les algorithmes proposés.

Il est clair à partir des figures 3.5 et 3.6 que les algorithmes proposés sont beaucoup plus rapides que les algorithmes optimaux. Ceci se conforme avec les équations de complexité asymptotique données dans les équations (3.26) et (3.32).

Cette partie présente la fonction de répartition de l'efficacité énergétique, c.-à-d., la probabilité d'avoir une efficacité énergétique inférieure à une certaine valeur. L'efficacité énergétique est une mesure qui a été largement étudiée dans la littérature (Mertikopoulos et Belmega, 2016; Buzzi *et al.*, 2016b) et elle est définie comme le rapport entre le débit de transmission total sur la puissance totale consommée. Dans cette partie, la puissance

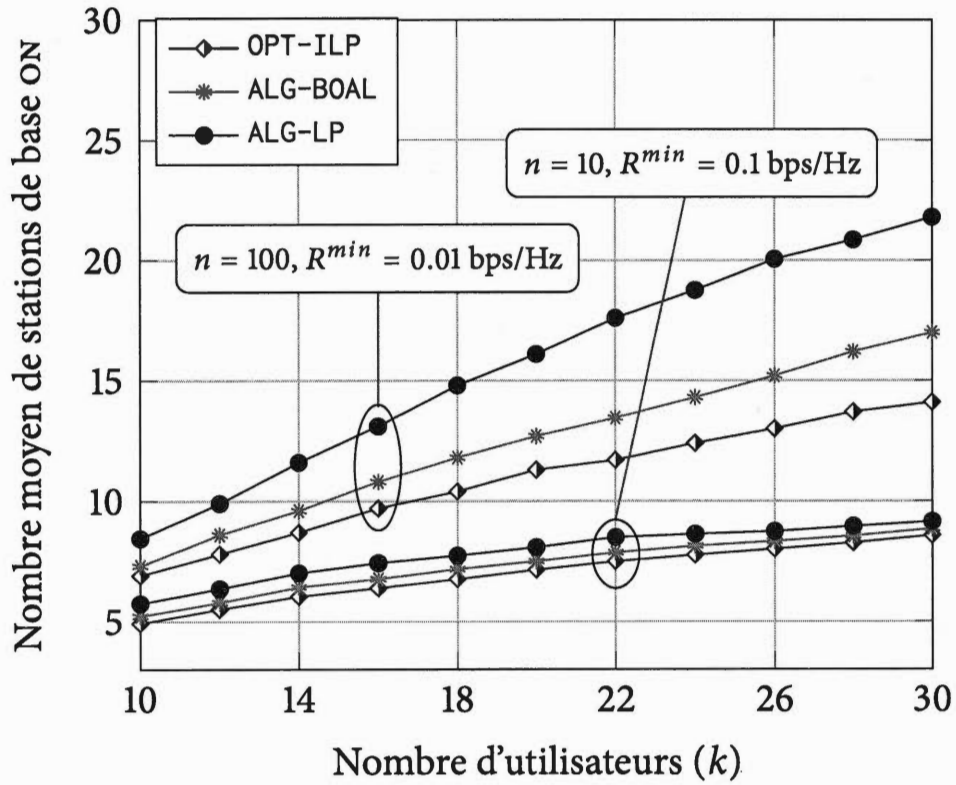


Figure 3.4 Nombre moyen de stations de base ON par tous les algorithmes en fonction de k pour BOAL.

statique p_b^s de la station de base b est égale à 20 dBm (Mertikopoulos et Belmega, 2016).

La figure 3.7 illustre la fonction de répartition de l'efficacité énergétique des différents algorithmes pour BOAH. Elle montre que OPT-BF donne la meilleure efficacité énergétique pour $k = 6$ et $n = 20$. Pour la même valeur que k et n , il est clair que ALG-BOAH donne une efficacité énergétique beaucoup plus grande que celle donnée par ALG-CB. En fait, pour une probabilité de 0,8, l'efficacité énergétique donnée par ALG-BOAH est environ 2,25 fois plus grand que celle donnée par ALG-CB. Même pour des valeurs plus grandes de k et n , ALG-BOAH donne toujours une meilleure efficacité énergétique, bien que l'écart de performance se réduise à environ 1,14.

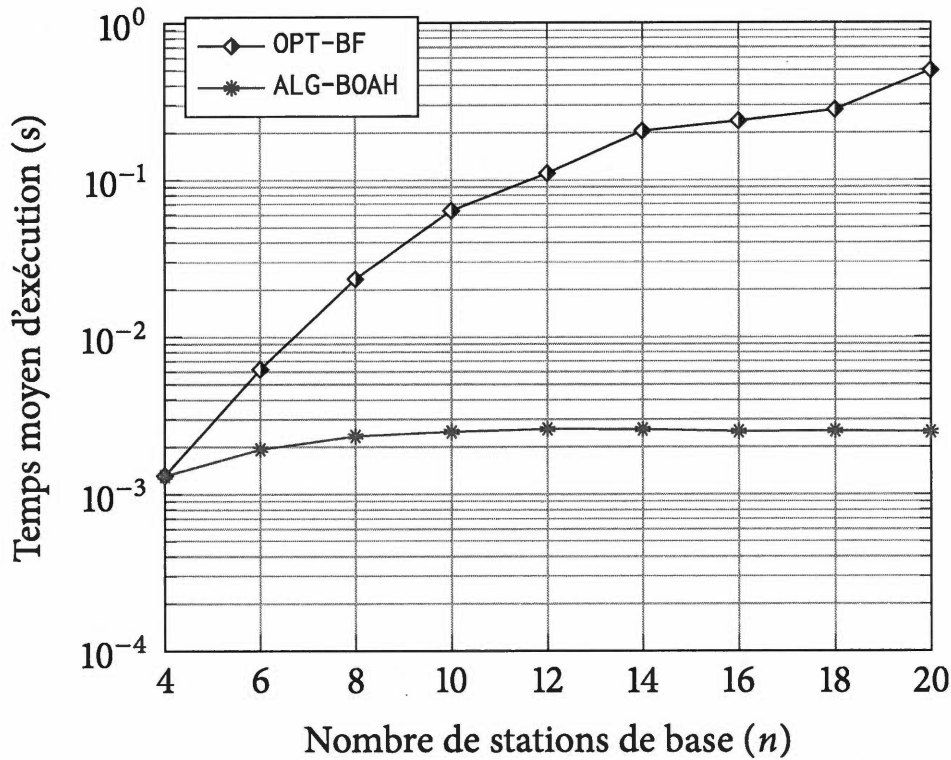


Figure 3.5 Temps d'exécution des algorithmes en fonction de n avec $k = 6$ pour BOAH.

La figure 3.8 compare les algorithmes proposés pour BOAL en termes d'efficacité énergétique. OPT-ILP donne la meilleure efficacité énergétique pour différentes valeurs de k et n . De plus, l'algorithme proposé ALG-BOAL donne une efficacité énergétique plus grande par rapport à celle donnée par ALG-LP. Par exemple, le gain en efficacité énergétique entre ALG-BOAL et ALG-LP est de 1,1 pour une probabilité de 0,8. Ce qui est plus intéressant est que l'efficacité énergétique obtenue par ALG-BOAL est très proche de celle obtenue par l'algorithme optimal lorsque n augmente.

À partir des deux figures 3.7 et 3.8, nous observons que si le nombre de stations de base et le nombre d'utilisateurs augmente, alors l'efficacité énergétique diminue car plus de stations de base sont activées et donc plus de puissance est consommée.

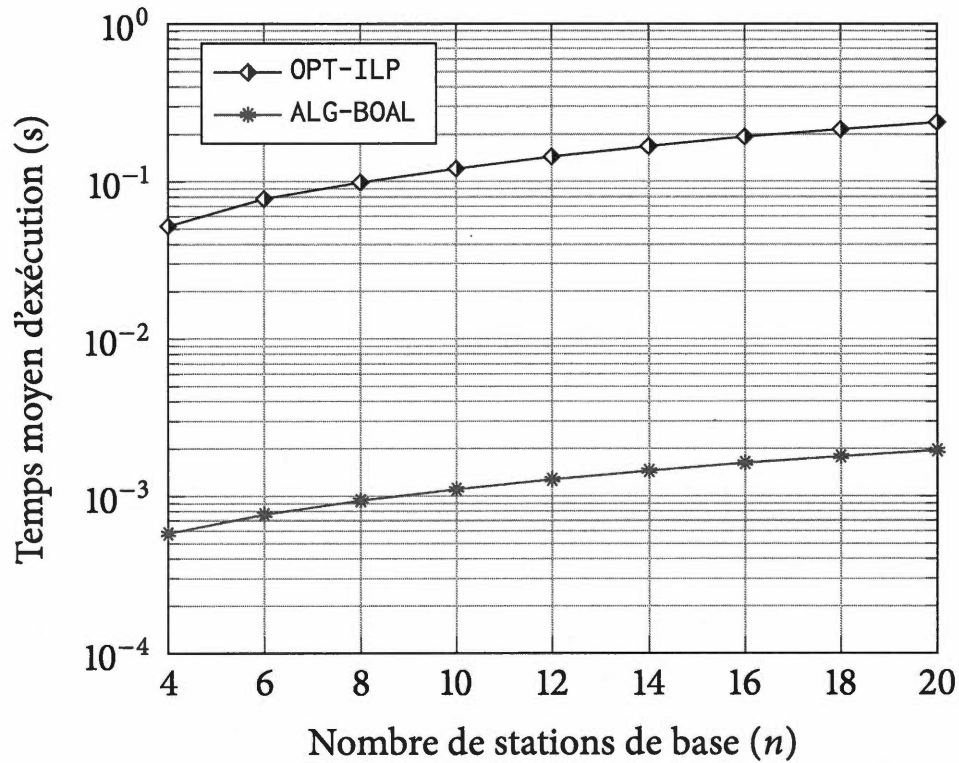


Figure 3.6 Temps d'exécution des algorithmes en fonction de n avec $k = 20$ pour BOAL.

Nous illustrons dans cette partie l'impact de la mobilité sur les performances des algorithmes proposés. En général, en raison de la mobilité des utilisateurs, les algorithmes doivent être réexécutés à des moments fixes afin de trouver les nouvelles associations et de s'adapter aux changements de topologie du réseau. Bien sûr, si la vitesse des utilisateurs est grande, il devrait exister une méthode spécifique de gestion de transfert inter-cellulaire (*handover* en anglais). Dans ce qui suit, nous évaluons le *temps de séjour* des algorithmes proposés. Ce temps est une métrique importante définie comme la durée moyenne d'association d'un utilisateur (Lin *et al.*, 2013).

Les simulations présentées dans la figure 3.9 considère le scénario suivant. Il existe $n = 10$ stations de base et $k = 6$ utilisateurs distribués aléatoirement selon une loi de probabilité uniforme. Nous nous concentrons sur un utilisateur u_0 qui se déplace le long

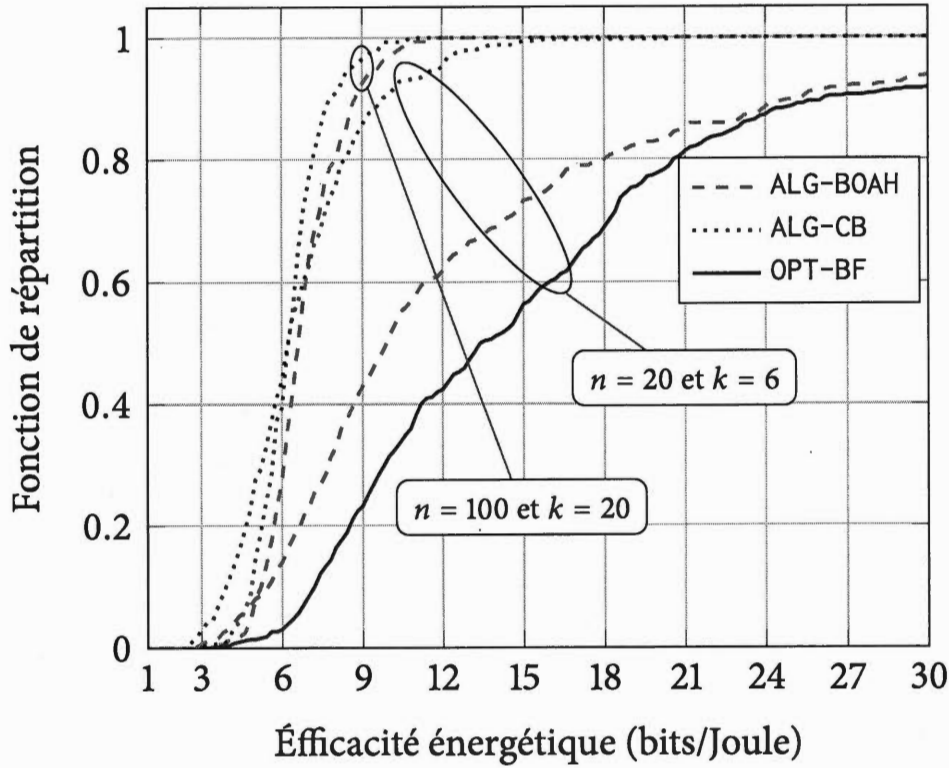


Figure 3.7 Fonction de répartition de l'efficacité énergétique par tous les algorithmes pour BOAH.

d'une ligne avec une vitesse v km/h. Quand il atteint la limite de la zone géographique, il inverse la direction. Nous considérons une période de 20 *slots*. Au début du premier *slot*, u_0 commence à se déplacer, les algorithmes sont exécutés et une solution d'association X est retournée. À chaque *slot* ultérieur, si l'association X change, alors le temps de séjour (exprimé en pourcentage) est évalué comme le rapport entre l'indice du *slot* précédent et la taille de toute la période, qui est 20 (si l'association change à la 11-ème *slot*, alors le pourcentage du temps de séjour est égal 10/20).

À partir de la figure 3.9, nous observons que lorsque la vitesse de u_0 est faible, comme le cas d'un modèle de mouvement piétonnier (Mertikopoulos et Belmega, 2016), les algorithmes retournent des valeurs de temps de séjour relativement élevées ; ce qui illustre

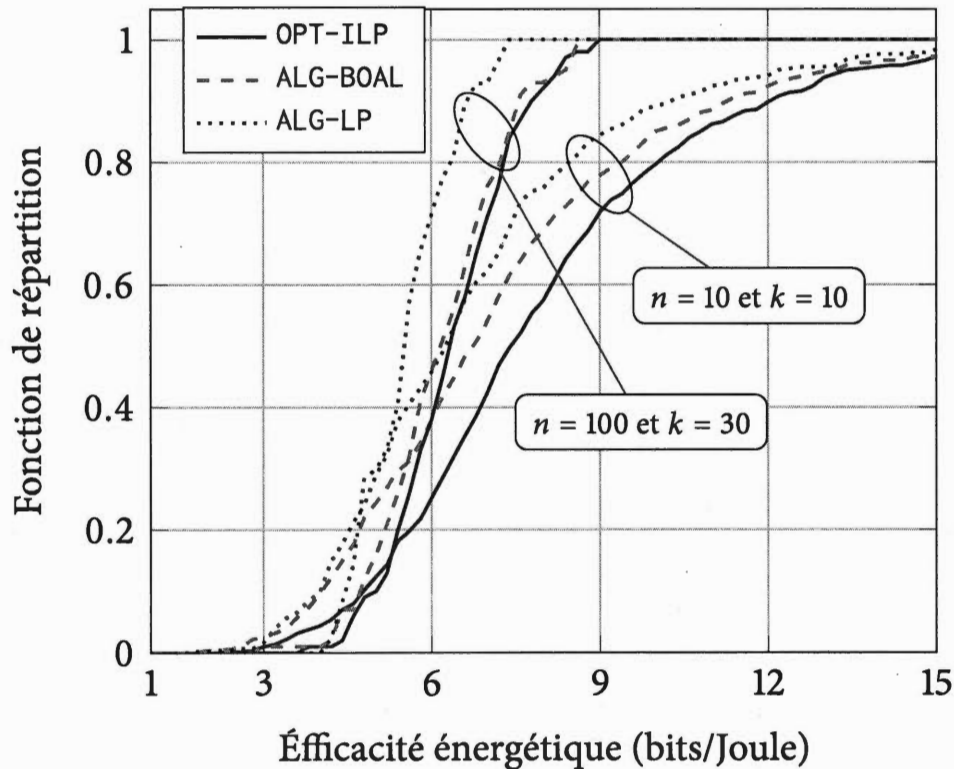


Figure 3.8 Fonction de répartition de l'efficacité énergétique par tous les algorithmes pour BOAL.

leur efficacité. Dès que la vitesse commence à augmenter, u_0 devrait changer sa station de base souvent, et donc une procédure de transfert intercellulaire est nécessaire; ce qui engendre une réduction au niveau du temps de séjour. Heureusement, nous remarquons que les algorithmes proposés se caractérisent par un temps de séjour élevé par rapport aux algorithmes optimaux. En outre, les temps de séjour pour les algorithmes de BOAL, sont plus élevés puisque les débits minima des utilisateurs sont faibles (évidemment plus les débits minima sont faibles plus les utilisateurs sont satisfaits).

Dans cette partie, nous étudions BOA dans le cas où les débits minima des utilisateurs sont hétérogènes. Ce problème s'appelle BOA avec débits minima mixte (BOAM). Pour résoudre BOAM, nous appliquons ALG-BOAL et nous le comparons à OPT-ILP

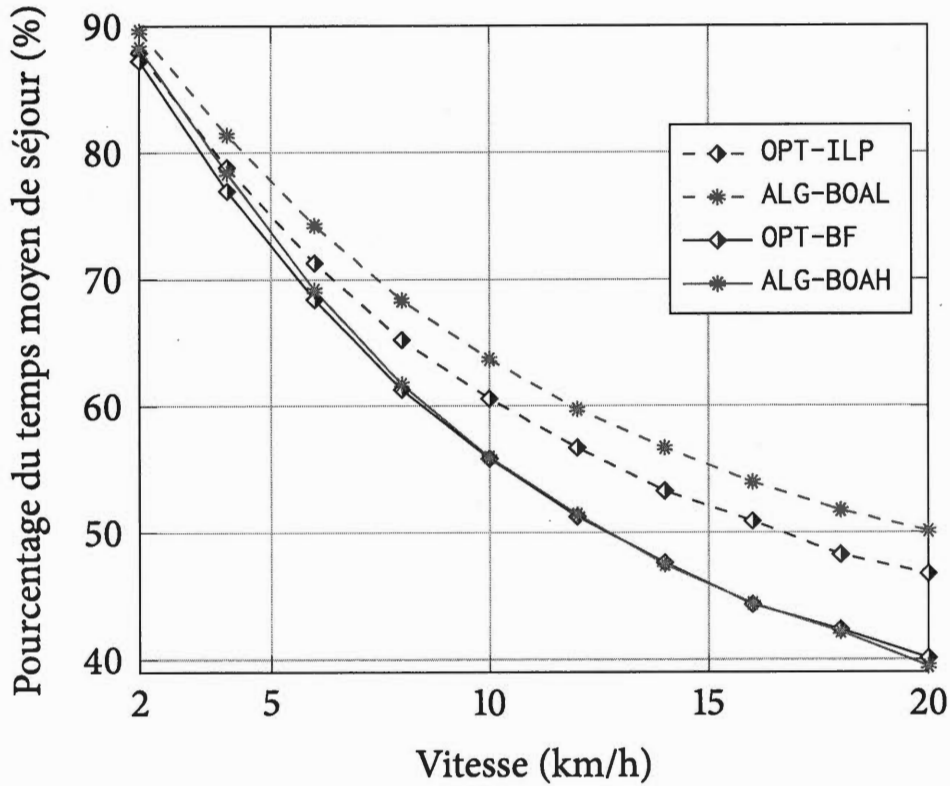


Figure 3.9 Temps de séjour moyen pour tous les algorithmes avec $k = 6$, $n = 10$ pour BOAH et BOAL.

et à ALG-LP. Nous considérons trois utilisateurs avec les débits hétérogènes suivants⁴ $R_1^{min} = R_2^{min} = 1$ bits/s/Hz et $R_3^{min} = 1,2$ bits/s/Hz et le reste des utilisateurs ont les débits minima faibles suivants $R_u^{min} = 0,01$ bits/s/Hz pour tout $u \neq 1, 2, 3$.

La figure 3.10 montre que l'algorithme ALG-BOAL reste toujours proche de l'optimal et meilleur que ALG-LP même avec des débits minima hétérogènes. En comparant les figures 3.4 et 3.10 pour $n = 100$, nous observons qu'il y a une légère augmentation du nombre de stations de base ON pour BOAM lorsque la valeur de k augmente. Ceci est attendu car il faut plus de stations de base pour satisfaire les utilisateurs avec des débits

4. Nous pouvons choisir n'importe quelles autres valeurs de débits minima.

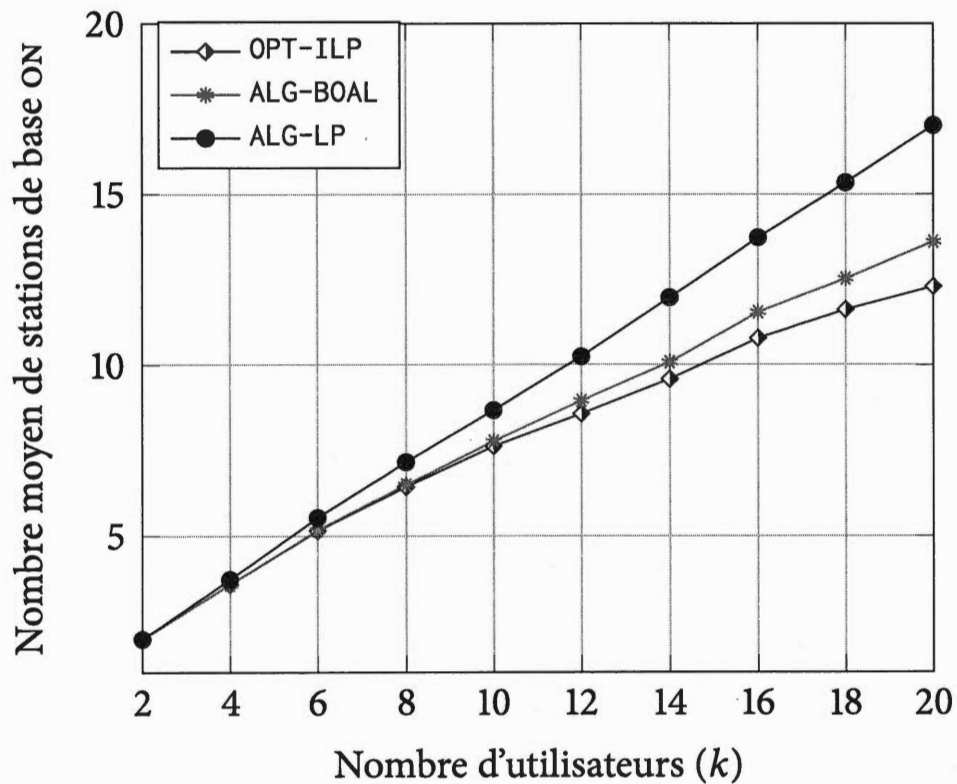


Figure 3.10 Nombre moyen de stations de base ON par tous les algorithmes en fonction de k avec $n = 100$ pour BOAM.

minima élevés.

3.8 Conclusions

Dans ce chapitre, nous avons étudié le problème d'activation des stations de base et d'association d'utilisateurs dans les réseaux DCNs (appelé BOA) afin de maximiser l'efficacité énergétique. Nous avons formulé BOA comme un programme mathématique en nombre entiers et nous avons montré qu'il est NP-difficile même de trouver une solution faisable. Nous avons analysé BOA dans deux cas : (i) BOA avec des débits minima élevés (BOAH) et (ii) BOA avec des débits minima faibles (BOAL). Quant à BOAH, nous avons

montré qu'il peut être découplé et ainsi nous avons pu résoudre les deux problèmes d'activation des stations de base et d'association des utilisateurs séparément. Ensuite, nous avons proposé un algorithme de force brute et nous avons considérablement amélioré sa complexité. Puis, nous avons proposé un algorithme heuristique polynomial qui permet de trouver très souvent des solutions faisables qui sont proches de l'optimal. Quant à BOAL, étant donnée que l'activation des stations de base et l'association des utilisateurs ne peuvent pas être découplées, nous avons simplifié le problème en supposant le pire cas d'interférence. Nous avons ainsi formulé BOAL comme un programme linéaire en nombres entiers. Pour le résoudre efficacement, nous avons proposé un algorithme d'approximation qui donne un facteur d'approximation (i) logarithmique lorsqu'il trouve une solution faisable ou (ii) constant, sinon. Nous avons montré l'efficacité de nos algorithmes proposés en faisant des simulations Monte Carlo.

CHAPITRE IV

ORDONNANCEMENT, ASSOCIATION DES UTILISATEURS ET ALLOCATION DE FRÉQUENCES SANS ET AVEC LA RÉCOLTE D'ÉNERGIE DANS LES SCÉNARIOS *OFFLINE* ET *ONLINE*

Dans les chapitres précédents, nous avons étudié le problème d'association des utilisateurs sans prendre en considération les ressources temporelles et fréquentielles. Bien que le chapitre 3 étudie l'association sur plusieurs *slots*, nous avons supposé quand-même que chaque utilisateur est servi pendant un seul *slot*. Cependant, lorsque la demande d'un utilisateur dépasse un *slot* ou lorsqu'il existe différentes fréquences disponibles, la station de base doit allouer ses ressources temporelles et fréquentielles aux utilisateurs afin de maximiser les performances de ses utilisateurs. Ce chapitre étudie le problème d'ordonnancement, d'association des utilisateurs et d'allocation de fréquences sans ou avec la récolte d'énergie dans deux scénarios différents; *offline* et *online*.

Ce travail a produit les quatre contributions (Mlika *et al.*, 2017d; Mlika *et al.*, 2018b; Mlika *et al.*, 2018d; Mlika *et al.*, 2019). Les résultats préliminaires qui sont présentés dans (Mlika *et al.*, 2018d) ne sont pas discutés dans ce chapitre.

4.1 Résumé

Dans un premier lieu, nous étudions le problème *offline* d'ordonnancement et d'association des utilisateurs sans récolte d'énergie. Le problème consiste à maximiser le nombre d'utilisateurs servis de telle sorte qu'ils respectent leurs dates d'arrivée ainsi que leurs dates limites. Nous étudions la NP-difficulté de ce problème et nous présentons un algorithme polynomial d'un cas spécial et un algorithme 2-approximation pour le cas général. Enfin, nous montrons via des simulations Monte Carlo que l'algorithme d'approximation proposé est quasi-optimal qu'il surpasse des algorithmes heuristiques simples.

Dans un second lieu, nous étudions le problème *offline* d'ordonnancement, d'association des utilisateurs et d'allocation de fréquences avec récolte d'énergie. L'utilisation de la récolte d'énergie permet d'assurer une communication verte. L'objectif de ce problème est de maximiser le nombre d'utilisateurs servis qui respectent leurs dates limites tout en leur allouant les fréquences disponibles et en respectant les contraintes de la récolte d'énergie. Premièrement, la NP-difficulté du problème est étudiée dans plusieurs cas. Ensuite, selon le cas, nous proposons des algorithmes optimaux, heuristiques ou d'approximation. Finalement, nous fournissons des simulations Monte Carlo afin d'illustrer les performances des algorithmes proposés.

En dernier lieu, nous étudions le problème *online* d'ordonnancement et d'association des utilisateurs avec récolte d'énergie. L'entrée du problème arrive *online*, c.-à-d. comme un flux de données et nous devons maximiser irrévocablement au fur et à mesure le nombre d'utilisateurs servis tout en respectant leurs dates limites et les contraintes énergétiques. Premièrement, nous proposons des algorithmes *online*s et nous montrons qu'il s'agit d'algorithmes compétitifs. Ensuite, nous proposons un algorithme d'apprentissage incrémental et nous montrons qu'il atteint un regret sous-linéaire au fil du temps. Fina-

lement, nous fournissons des simulations Monte Carlo afin d'illustrer les performances des algorithmes proposés.

4.2 Introduction

Les exigences clés de la cinquième génération (5G) des réseaux cellulaires peuvent être regroupées en deux catégories : une efficacité spectrale élevée et une latence ultra-faible (Andrews *et al.*, 2014; Damnjanovic *et al.*, 2011). Les déploiements initiaux de la 5G se concentrent sur les applications améliorées de haut débit mobile (*enhanced Mobile BroadBand (eMBB)*) où l'efficacité spectrale est l'un des indicateurs de performance clés les plus importants (Liu *et al.*, 2017). Évidemment, comme montré précédemment, l'utilisation d'une seule fréquence augmente l'efficacité spectrale à condition que l'interférence soit bien gérée. Nous avons montré dans les chapitres précédents que l'allocation des ressources spatiales est un moyen efficace pour améliorer l'efficacité spectrale. Néanmoins, l'allocation des ressources temporelles et fréquentielles est aussi nécessaire. Dans ce chapitre, nous nous concentrons sur le problème d'allocation des ressources spatiales, temporelles et fréquentielles sous des contraintes d'énergie et de délai. Ce problème est appelé *problème d'allocation des ressources* (PAR). Dans PAR, chaque utilisateur demande de télécharger des données après une date d'arrivée et avant une date limite. Afin de maximiser l'utilisation d'énergie verte, les stations de base utilisent la technologie de la récolte d'énergie. L'objectif de PAR est d'associer et d'ordonnancer le plus grand nombre d'utilisateurs tout en leur allouant les fréquences disponibles et en respectant les contraintes imposées par les dates d'arrivée, dates limites et l'énergie récoltée. Nous divisons l'étude de PAR en trois problèmes : (i) *ordonnancement et association avec dates limites* (OAD), (ii) *ordonnancement, association et fréquence avec énergie et dates limites* (OAFED), et (iii) *ordonnancement et association avec énergie et dates limites online* (OAED-ON). Dans OAD, le problème consiste à ordonnancer et associer, d'une manière *offline*, le maximum nombre d'utilisateurs en respectant leurs dates d'arrivée et dates li-

mites. Dans OAFED, le problème consiste à ordonnancer et associer, d'une manière *offline*, le maximum nombre d'utilisateurs en leur allouant les fréquences disponibles et en respectant leurs dates limites et les contraintes de l'énergie récoltée. Dans OAED-ON, le problème consiste à ordonnancer et associer, d'une manière *online*, le maximum nombre d'utilisateurs en respectant leurs dates limites et les contraintes de l'énergie récoltée.

4.2.1 Contributions

Les contributions de ce chapitre sont les suivantes :

- Nous formulons OAD comme un programme linéaire en nombres entiers.
- Nous étudions la NP-difficulté d'OAD dans différents cas.
- Nous étudions OAD dans un cas particulier et nous fournissons un algorithme polynomial optimal.
- Nous proposons un algorithme 2-approximation pour OAD et nous le comparons avec l'optimal ainsi qu'avec des heuristiques simples.
- Nous formulons OAFED comme un programme linéaire en nombres entiers.
- Nous étudions la NP-difficulté d'OAFED dans différents cas.
- Nous proposons des algorithmes polynomiaux optimaux, des algorithmes heuristiques, ainsi que des algorithmes d'approximation et nous les comparons ensemble.
- Nous formulons un cas spécial d'OAED-ON comme un programme linéaire en nombres entiers.
- Nous proposons des algorithmes *online*s et nous montrons qu'il s'agit d'algorithmes compétitifs pour OAED-ON.
- Nous proposons un algorithme d'apprentissage incrémental et nous montrons qu'il atteint un regret sous-linéaire au fil du temps pour OAED-ON.

4.2.2 Organisation

Le reste de ce chapitre est organisé comme suit. La section 4.3 présente et analyse une revue de littérature. La section 4.4 présente le modèle de système général de PAR. La section 4.5 présente OAD, elle décrit les solutions proposées et présente les résultats de simulation. La section 4.6 présente OAFED, elle décrit les solutions proposées et présente les résultats de simulation. La section 4.7 présente OAED-ON ainsi que les modifications nécessaires du modèle de système, elle discute les solutions proposées et illustre leurs performances via des simulations. Finalement, la section 4.8 conclut le chapitre.

4.3 Revue de littérature

Les travaux de recherche les plus importants liés à PAR sont discutés ci-après. Nous commençons d'abord par les travaux de recherche liés aux OAD et OAFED, ensuite nous présentons ceux liés à OAED-ON.

Dans (Zhuang *et al.*, 2017b), les auteurs proposent une approche d'allocation de fréquences et d'association d'utilisateurs dans les réseaux hétérogènes afin de minimiser le temps de transmission moyen des paquets. Leur approche se base sur un algorithme itératif qui résout un problème d'optimisation convexe et un problème de coloration d'hypergraphe. Des problèmes similaires à (Zhuang *et al.*, 2017b) sont étudiés dans (Zhuang *et al.*, 2016; Zhuang *et al.*, 2018). Dans (Zhao *et al.*, 2017), les auteurs étudient le problème d'allocation de fréquences et de contrôle de puissance dans les réseaux cellulaires à accès multiple non-orthogonal. Ils utilisent la théorie des jeux ainsi que l'optimisation séquentielle convexe. Les auteurs de (Lin *et al.*, 2015) étudient le problème d'association des utilisateurs avec et sans allocation de fréquences. En se basant sur la géométrie stochastique, ils calculent les facteurs analytiques d'association et d'allocation de fréquences. Les travaux dans (Zhuang *et al.*, 2017b; Zhuang *et al.*, 2016; Zhuang *et al.*, 2018; Zhao *et al.*,

2017; Lin *et al.*, 2015) étudient le problème d'allocation de fréquences ou celui d'association d'utilisateurs sans tenir compte du problème d'ordonnancement des utilisateurs ni de contraintes de récolte d'énergie. Dans (Huang *et al.*, 2016), les auteurs étudient l'ordonnancement multidiffusion des utilisateurs avec des dates limites dans les réseaux cellulaires. Dans (Deshmukh et Vaze, 2016), les auteurs étudient l'ordonnancement des paquets avec des dates limites communes afin de minimiser la consommation d'énergie sans et avec récolte d'énergie. Les travaux dans (Huang *et al.*, 2016; Deshmukh et Vaze, 2016) traitent exclusivement le problème d'ordonnancement sans aucune référence au problème d'association des utilisateurs. Dans (Krishnasamy et Shakkottai, 2015), l'allocation de fréquences et l'ordonnancement d'utilisateurs avec des dates limites communes est étudié dans un réseau DCN avec des communications appareil-à-appareil (*device-to-device* en anglais). Dans (Wang et Chen, 2017), les auteurs considèrent le problème d'ordonnancement en temps réel des paquets dans les réseaux *long term evolution-advanced* (LTE-A). Ils utilisent la méthode d'*almost blank subframe* (ABS) afin de proposer un algorithme d'ordonnancement efficace. La plupart des travaux de recherche précédents supposent des modèles de système différents par rapport de ceux de OAD et OAFED et ne fournissent pas des études de NP-difficulté ni d'algorithmes d'approximation.

Dans (Cygan *et al.*, 2016) le problème du sac à dos *online* (*online knapsack problem* (OKP)) est étudié. Dans OKP, un ensemble d'objets arrivent dans chaque période au fur et à mesure, où chaque objet a un poids et une valeur. Il y a un sac à dos de capacité connue. L'objectif est de maximiser irrévocablement la valeur globale des objets placés dans le sac à dos tout en respectant la capacité du sac dans chaque période. Une version d'OAED-ON peut être modélisée comme un OKP comme suit. Les utilisateurs correspondent aux objets, une station de base correspond au sac à dos et la quantité d'énergie disponible correspond à la capacité du sac. En général, OKP n'admet pas d'algorithmes compétitif (Cygan *et al.*, 2016). Par conséquent, de nombreuses hypothèses sont utilisées dans la littérature afin de concevoir des algorithmes compétitifs. OKP devient en-

core plus difficile lorsque la capacité du sac est variable. Un tel problème est connu dans la littérature comme *OKP with incremental capacity (OKP-IC)* (Thielen *et al.*, 2016). Dans OKP-IC, la capacité du sac augmente par une constante connue dans chaque période. Les auteurs de (Thielen *et al.*, 2016) proposent des algorithmes déterministes et randomisés compétitifs. OAED-ON est fondamentalement différent de OKP et de OKP-IC car les utilisateurs ont des dates limites et l'arrivée d'énergie est supposée inconnue et imprévisible. D'autre part, une version d'OAED-ON est liée au problème *online* de maximisation du débit de transmission (*online throughput maximization problem (OTMP)*) sous les contraintes de variation des canaux sans fil et de puissance. Dans (Buchbinder *et al.*, 2012), les auteurs étudient OTMP, le modélisent avec OKP et proposent des algorithmes compétitifs. Dans (Vaze *et al.*, 2014), les auteurs considèrent OTMP avec des contraintes de récolte d'énergie et proposent des algorithmes compétitifs. Dans (Shen *et al.*, 2016), les auteurs étudient le problème de la gestion de la mobilité dans les réseaux DCNs afin d'optimiser la consommation d'énergie. Ils modélisent leur problème en utilisant le problème de bandit manchot et proposent des algorithmes minimisant le regret. Nous remarquons qu'OAED-ON n'a pas été étudié dans la littérature.

Dans ce chapitre, nous comblons les lacunes de recherche suivantes (i) nous étudions le problème conjoint d'ordonnancement, d'association des utilisateurs et d'allocation de fréquences sans et avec récolte d'énergie, (ii) nous étudions la NP-difficulté de ce problème dans différents cas, et (iii) nous proposons des algorithmes efficaces *onlines* et *offline* offrant des garanties de performances dans les cas les plus défavorables.

4.4 Modèle de système

Nous considérons un réseau DCN composé de n stations de base qui utilisent la récolte d'énergie et qui sont dénotées par l'ensemble $\mathcal{B} \triangleq \{1, 2, \dots, n\}$. La bande passante disponible est divisée en un ensemble de ℓ fréquences orthogonales dénotées par

$\mathcal{F} \triangleq \{1, 2, \dots, \ell\}$. Les stations de base utilisent ces fréquences pour communiquer sur le lien descendant avec k utilisateurs dénotés par l'ensemble $\mathcal{U} \triangleq \{1, 2, \dots, k\}$. La puissance de transmission de la station de base b sur n'importe quelle fréquence est dénotée par p_b comme dans (Zhuang *et al.*, 2017b). Le temps est divisé en des trames de m *slots* chacune où chaque *slot* est d'une durée de τ secondes. PAR est résolu au début de chaque trame. Soit $\mathcal{S} \triangleq \{1, 2, \dots, m\}$ l'ensemble de *slots* d'une trame donnée. Chaque utilisateur u a une requête de données représentée par le triplet (w_u, r_u, d_u) où w_u est sa taille en bits, r_u est sa date d'arrivée et d_u est sa date limite. Pour simplifier les notations, nous appelons r_u et d_u l'arrivée et la date limite de u , respectivement. Les stations de base peuvent utiliser exclusivement la récolte d'énergie. Dans ce chapitre, nous considérons deux cas : sans récolte d'énergie et avec récolte d'énergie. Dans le cas sans récolte d'énergie, les stations de base utilisent l'électricité domestique comme une source d'énergie. Dans cette partie, nous présentons le modèle de système dans le cas général et nous le modifions (si nécessaire) au fur et à mesure. À chaque *slot* $s \in \mathcal{S}$, la quantité d'énergie disponible à la station de base b est désignée par ¹ $E_{b,s}$. Cette quantité d'énergie est stockée dans la batterie de b qui est supposée avoir une grande capacité (Shan *et al.*, 2015) ². Dans la suite de ce chapitre, nous normalisons $E_{b,s}$ par p_b , c.-à-d., $A_{b,s} \triangleq E_{b,s}/p_b$. Le terme $A_{b,s}$ représente le nombre minimal ³ de *slots* que la station de base b peut utiliser en transmettant avec la puissance p_b à partir du *slot* s (c.-à-d., au moins $A_{b,s}$ *slots* sont disponibles à b à partir de s). En se basant sur cette normalisation, l'énergie et le temps sont traités de manière équivalente, autrement dit, si l'énergie est disponible pendant le *slot* s , alors il existe au moins un *slot* disponible qui peut être utilisé par un utilisateur à partir de $s' \geq s$. L'arrivée de l'énergie $E_{b,s}$ (et donc le nombre de *slots* $A_{b,s}$) est arbitraire et peut prendre n'importe

1. C'est la quantité d'énergie récoltée jusqu'à présent et disponible au *slot* s .

2. L'hypothèse d'une batterie de grande capacité est utilisée pour simplifier l'analyse seulement. Cependant, nous considérons une capacité limitée dans le scénario *online* au niveau de la section 4.7.

3. Puisqu'il peut y avoir de l'énergie résiduelle provenant de *slots* précédents.

quelle valeur non négative. Ainsi, nous ne supposons pas que sa distribution suit des lois particulières, autrement dit, nous considérons le cas non stochastique général.

Lorsque l'utilisateur $u \in \mathcal{U}$ est associé à la station de base $b \in \mathcal{B}$ sur la fréquence $f \in \mathcal{F}$, le rapport SINR est donné par :

$$SINR_{b \rightarrow u}^f = \frac{p_b g_{b \rightarrow u}^f}{\sigma^2 + \sum_{b' \in \mathcal{B} \setminus \{b\}} p_{b'} g_{b' \rightarrow u}^f}, \quad (4.1)$$

où σ^2 est la puissance du bruit AWGN et $g_{b \rightarrow u}^f$ est le gain du canal sans fil entre u et b sur f , prend en compte l'évanouissement à grande échelle (Zhuang *et al.*, 2016) et donc ne varie pas rapidement en fonction du temps. Autrement dit, $g_{b \rightarrow u}^f$, les rapports SINR ainsi que les débits de transmission sont supposés constants pendant la période S de m slots mais ils varient après chaque période de m slots d'une manière indépendante.

À partir de (4.1), le débit de transmission $R_{b \rightarrow u}^f$ (en bits/s/Hz) entre u et b sur f peut être calculé comme :

$$R_{b \rightarrow u}^f = \log_2(1 + SINR_{b \rightarrow u}^f). \quad (4.2)$$

L'utilisateur u est satisfait lorsqu'il est associé à une station de base b sur une fréquence f en lui allouant $v_{b \rightarrow u}^f$ slots, où $v_{b \rightarrow u}^f$ est défini comme :

$$v_{b \rightarrow u}^f \triangleq \left\lceil \frac{w_u \ell}{\tau B R_{b \rightarrow u}^f} \right\rceil, \quad (4.3)$$

où B est la bande passante totale en Hz. En fait, l'égalité (4.3) représente un rapport entre la taille d'une requête de l'utilisateur u , w_u , et le nombre de bits obtenus avec la station de base b sur la fréquence f . Le dénominateur de ce rapport est égal à : $\tau B R_{b \rightarrow u}^f / \ell$, où $B R_{b \rightarrow u}^f / \ell$ représente le débit en bits/s. L'utilisateur u doit télécharger ses données de w_u

bits en s'associant à une station de base b sur une fréquence f pour une période de $v_{b \rightarrow u}^f$ slots et ceci après sa date d'arrivée r_u et avant sa date limite d_u et lorsque l'énergie est disponible à b . Dans le reste de ce chapitre, lorsqu'il n'y a qu'une seule station de base (ou une seule fréquence) dans le réseau, nous supprimons l'indice b (ou l'indice f) des différentes notations. La figure 4.1 donne un exemple de modèle de système considéré.

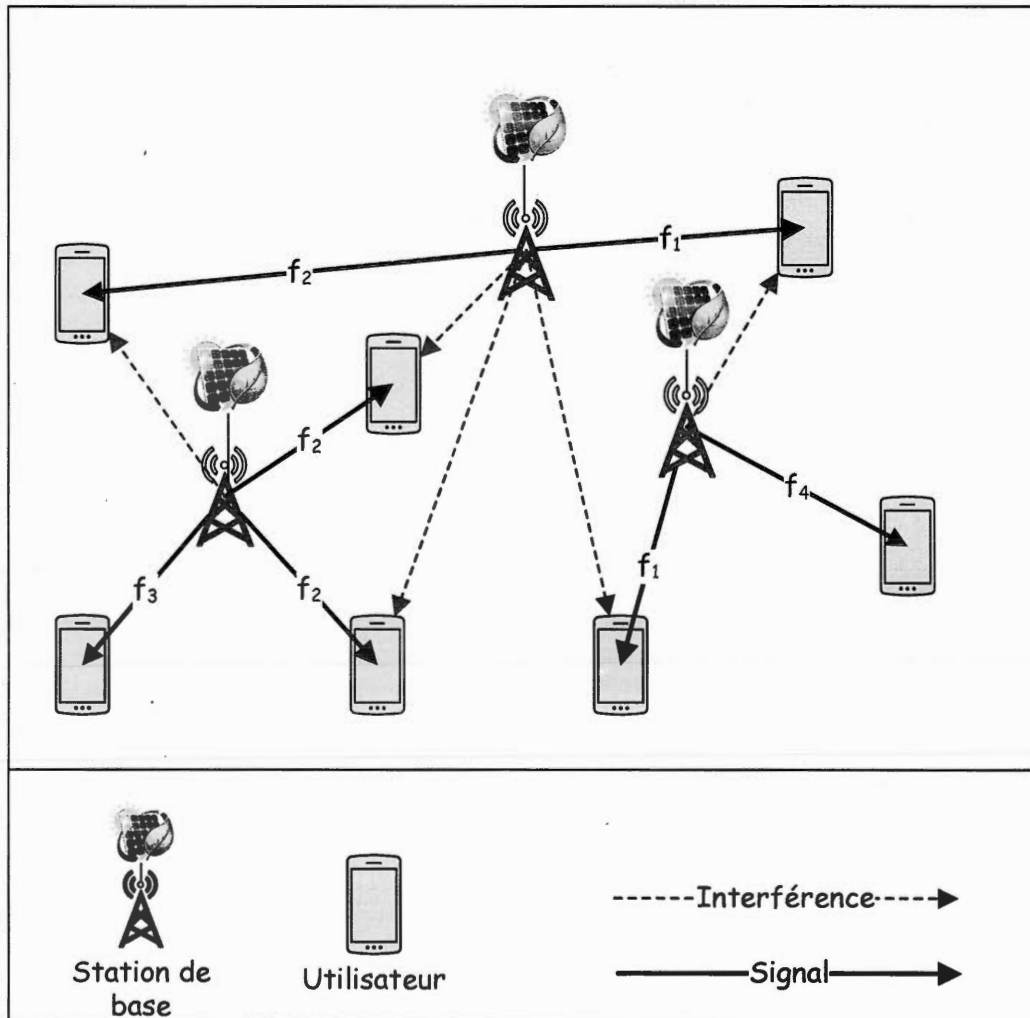


Figure 4.1 Un exemple du modèle de système où $\ell = 4$ et $f_i = i$.

L'objectif de PAR est de maximiser le nombre d'utilisateurs servis pendant l'ensemble

de m slots tout en (i) leur allouant les fréquences disponibles, (ii) satisfaisant leurs demandes, leurs dates d'arrivée et leurs dates limites, et (iii) respectant la quantité d'énergie disponible.

4.5 Le problème OAD

Dans cette section, nous considérons PAR dans le cas d'ordonnancement et d'association des utilisateurs sans récolte d'énergie (c.-à-d., OAD). Autrement dit, il y a une seule fréquence ($\mathcal{F} = \{1\}$) et l'énergie récoltée par les stations de base est suffisamment grande.

4.5.1 Formulation du problème

Afin de formuler OAD comme un programme linéaire en nombres entiers, nous introduisons les variables binaires suivantes :

$$x_{b \rightarrow u}^s \triangleq \begin{cases} 1, & \text{si } u \text{ est associé à } b \text{ pendant } [s, s + v_{b \rightarrow u}], \\ 0, & \text{sinon} \end{cases} \quad (4.4)$$

et $\mathbf{X} \triangleq [x_{b \rightarrow u}^s]$ représente la notation matricielle des variables $x_{b \rightarrow u}^s$.

L'objectif d'OAD est de maximiser le nombre d'utilisateurs associés pendant les m slots de telle sorte que leurs dates d'arrivée et dates limites soient respectées et que leurs demandes soient satisfaites. Par conséquent, OAD peut être formulé comme suit :

$$\text{maximiser} \quad \sum_{u \in \mathcal{U}} \sum_{b \in \mathcal{B}} \sum_{s=r_u}^{d_u - v_{b \rightarrow u}} x_{b \rightarrow u}^s \quad (4.5a)$$

$$\text{sous les contraintes} \quad \sum_{b \in \mathcal{B}} \sum_{s=r_u}^{d_u - v_{b \rightarrow u}} x_{b \rightarrow u}^s \leq 1, \quad \forall u \in \mathcal{U} \quad (4.5b)$$

$$\sum_{u \in \mathcal{U}} \sum_{s'=s-v_{b \rightarrow u}+1}^s x_{b \rightarrow u}^{s'} \leq 1, \quad \forall (b, s) \in \mathcal{B} \times \mathcal{S} \quad (4.5c)$$

$$x_{b \rightarrow u}^s = 0, \quad \forall (u, b) \in \mathcal{U} \times \mathcal{B}, s \notin [r_u, d_u] \quad (4.5d)$$

$$x_{b \rightarrow u}^s \in \{0, 1\}, \quad \forall (u, b, s) \in \mathcal{U} \times \mathcal{B} \times \mathcal{S}. \quad (4.5e)$$

La fonction objectif (4.5a) maximise le nombre d'utilisateurs associés. Les contraintes (4.5b) garantissent que l'utilisateur u ne peut pas être associé à plus qu'une station de base à n'importe quel *slot*. Les contraintes (4.5c) garantissent que les utilisateurs associés à la station de base b ne doivent pas interférer au *slot* s . Il est important de définir que deux utilisateurs interfèrent pendant le *slot* s s'ils sont associés à la même station de base pendant ce *slot*. Les contraintes (4.5d) garantissent que l'utilisateur u ne peut pas être associé à la station de base b pendant le *slot* $s \notin [r_u, d_u]$. Finalement, les contraintes (4.5e) garantissent que les variables $x_{b \rightarrow u}^s$ sont binaires pour tout $(u, b, s) \in \mathcal{U} \times \mathcal{B} \times \mathcal{S}$. En utilisant des solveurs de programmes linéaires, nous pouvons obtenir un algorithme optimal qui résoud (4.5) en se basant sur la technique de *branch and bound*. Cet algorithme optimal est noté OPT-OAD.

Dans le reste de ce chapitre, nous utilisons la définition suivante d'un ordonnanceur.

Définition 4.5.1 (Un ordonnanceur) :

Un ordonnanceur $\Sigma = [\sigma_1, \sigma_2, \dots, \sigma_m]$ est un vecteur de taille m qui associe à chaque utilisateur $u \in \{1, 2, \dots, k\}$ un *slot* $s \in \{1, 2, \dots, m\}$ ⁴. Autrement dit, si $\sigma_s = u$ alors u est servi pendant s (le *slot* s est appelé un *slot* occupé) et si $\sigma_s = 0$, alors le *slot* s est inoccupé.

La figure 4.2 illustre un exemple d'une solution faisable à une instance arbitraire de (4.5). Dans cette figure, nous considérons une solution avec $m = 5$, $n = 2$, $k = 5$, $r_u = 0$ pour tout u , $\mathbf{d} = [d_1, d_2, d_3, d_4, d_5] = [3, 4, 5, 4, 5]$ et les *slots* requis par les utilisateurs sont : $v_{1 \rightarrow 1} = 2$, $v_{1 \rightarrow 2} = 1$, $v_{1 \rightarrow 3} = 1$, $v_{1 \rightarrow 4} = 4$, $v_{1 \rightarrow 5} = 4$, $v_{2 \rightarrow 1} = 4$, $v_{2 \rightarrow 2} = 4$, $v_{2 \rightarrow 3} = 4$, $v_{2 \rightarrow 4} = 3$, et $v_{2 \rightarrow 5} = 2$. On voit que les utilisateurs servis, 1, 2, 3, et 4 sont satisfaits.

4. Par convention, on utilise le *slot* 0 pour désigner un *slot* vide.

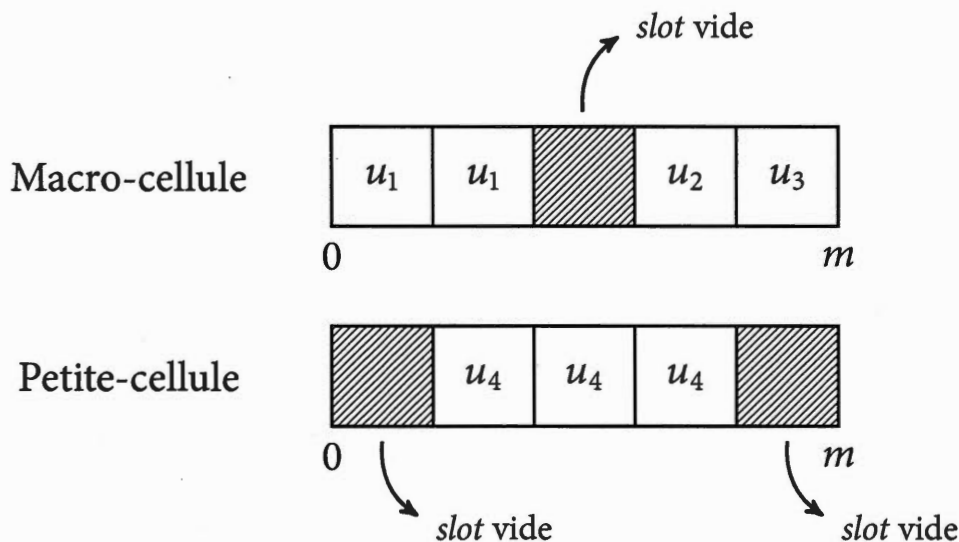


Figure 4.2 Un exemple d'une solution faisable à (4.5) qui est constituée de deux ordonnancements; un pour une macro-cellule et un pour une petite cellule.

4.5.2 NP-difficulté et solutions proposées

4.5.2.1 Le cas d'une seule station de base

Supposons que $\mathcal{B} = \{1\}$ et notons OAD par OAD_1 . Ce cas particulier mérite d'être étudié afin de nous aider à résoudre le problème dans le cas général efficacement. D'abord, nous étudions la NP-difficulté de OAD_1 .

Lemme 4.5.2 *OAD_1 est NP-difficile lorsque les dates d'arrivée sont arbitraires mais appartient à la classe de complexité P lorsque les dates d'arrivée sont identiques.*

Démonstration : supposons d'abord que les dates d'arrivée des utilisateurs sont arbitraires. Dans ce cas, OAD_1 devient équivalent au problème bien connu d'ordonnement avec intervalles (*sequencing with intervals* en anglais) (Garey et Johnson, 1979). Par conséquent, nous pouvons réduire, en temps polynomial, le problème 3-PARTITION à OAD_1 en utilisant la réduction de 3-PARTITION à l'ordonnement avec intervalles.

La preuve complète est donnée dans (Garey et Johnson, 1979). Maintenant, supposons que les dates d'arrivée des utilisateurs sont identiques, soient $r_u = 0$ pour tout u . L'algorithme « première date limite, premier servi » (ou en anglais *Earliest Deadline First (EDF)*) peut être modifié à mEDF (Pinedo, 2016) pour donner une solution optimale à OAD_1 en $\mathcal{O}(k \log k)$. mEDF est un algorithme itératif qui trie les utilisateurs par ordre croissant des dates limites et ordonnance un utilisateur à chaque itération. Dans n'importe quelle itération, si un utilisateur ordonnancé ne respecte pas sa date limite, alors mEDF enlève l'utilisateur qui requiert le plus grand nombre de *slots* de l'ensemble des utilisateurs ordonnancés. À chaque itération, l'algorithme mEDF maintient un ensemble d'utilisateurs de cardinalité maximale qui respectent leurs dates limites et ont le plus petit nombre de *slots* requis. Cela mène à une solution optimale à OAD_1 en $\mathcal{O}(k \log k)$ où $k \log k$ est dû principalement au tri. Donc OAD_1 appartient à la classe de complexité P. Finalement, OAD_1 est NP-difficile lorsque les dates d'arrivée sont arbitraires, mais appartient à la classe de complexité P lorsque les dates d'arrivée sont identiques. Ceci prouve le lemme. ■

4.5.2.2 Le cas de plusieurs stations de base

Lorsque le réseau est composé de plusieurs stations de base, OAD devient NP-difficile même si les dates d'arrivée des utilisateurs sont identiques comme le montre le lemme suivant.

Théorème 4.5.3 *OAD est NP-difficile.*

Démonstration : soit $r_u = 0$ et $d_u = m$ pour tout $u \in \mathcal{U}$. Dans ce cas, OAD devient équivalent au problème d'affectation généralisée (ou en anglais *generalized assignment problem (GAP)*) (Chekuri et Khanna, 2005). Dans GAP, l'objectif est de maximiser le nombre d'objets mis dans des sacs tel que la capacité de chaque sac soit respectée où chaque objet a un poids qui dépend de chaque sac. Par analogie, si les utilisateurs représentent les

objets (les poids sont le nombre de *slots* requis par les utilisateurs) et les stations de base représentent les sacs de capacité m chacun, alors OAD et GAP sont équivalents. Cette réduction est trivialement polynomiale. Puisque GAP est NP-difficile, alors OAD l'est aussi. Ceci prouve le théorème. ■

Afin de résoudre OAD efficacement, nous proposons un algorithme d'approximation ainsi que trois algorithmes heuristiques adaptés à OAD. Nous proposons ces trois heuristiques puisqu'elles sont bien connues pour résoudre le problème d'ordonnancement avec intervalles (Garey et Johnson, 1979).

Un algorithme d'approximation

L'algorithme d'approximation qui résout OAD est nommé ALG-OAD. L'idée principale de cet algorithme est d'itérer sur l'ensemble des stations de base et d'associer, à chaque itération, les utilisateurs qui ont les plus petites dates limites et qui finissent le plus tôt possible lorsqu'ils sont ordonnancés (bien sûr, il faut considérer les utilisateurs déjà arrivés). Le pseudo-code de ALG-OAD est donné dans l'algorithme 7.

En détails, ALG-OAD fonctionne comme suit. Il itère l'ensemble des stations de base pendant les m *slots*. Pour chaque station de base b et pour chaque *slot* s , il trouve l'ensemble des utilisateurs admissibles qui sont définis par (i) ceux dont la date d'arrivée est inférieure ou égale à s et (ii) ceux qu'une fois ordonnancés pendant s , terminent avant leurs dates limites. Parmi ces utilisateurs, ALG-OAD associe celui qui termine en premier à b (c.-à-d., qui a le plus petit nombre de *slots* requis). Dans le cas où deux utilisateurs ou plus finissent en même temps, ALG-OAD choisit celui qui a la date limite la plus petite⁵.

Théorème 4.5.4 *ALG-OAD est un algorithme 2-approximation pour OAD.*

5. Le choix de l'algorithme est arbitraire dans tout autre cas de bris d'égalité.

Algorithme 7 L'algorithme ALG-OAD

```

1:  $\mathcal{L} \leftarrow \emptyset$ 
2:  $x_{b \rightarrow u}^s \leftarrow 0$ 
3: pour  $b \leftarrow 1$  à  $n$  faire
4:    $s \leftarrow 0$ 
5:   tant que  $s \leq m - 1$  faire
6:      $\mathcal{A} \leftarrow \emptyset$ 
7:     pour  $u$  dans  $\mathcal{U} \setminus \mathcal{L}$  faire
8:        $m_u \leftarrow \max(r_u, s)$ 
9:       si  $m_u + v_{b \rightarrow u} \leq d_u$  alors
10:         $\mathcal{A} \leftarrow \mathcal{A} \cup \{u\}$ 
11:      fin si
12:    fin pour
13:     $i \leftarrow \arg \min \{m_u + v_{b \rightarrow u} : u \in \mathcal{A}\}$ 
14:     $x_{b \rightarrow i}^{m_i} \leftarrow 1$ 
15:     $\mathcal{L} \leftarrow \mathcal{L} \cup \{i\}$ 
16:     $s \leftarrow m_i + v_{b \rightarrow i}$ 
17:  fin tant que
18: fin pour
19: retourner  $\mathbf{X} = [x_{b \rightarrow u}^s]$ 

```

Démonstration : soit $\mathcal{U}^*$ (resp. $\mathcal{U}$) l'ensemble des utilisateurs associés par OPT-OAD (resp. ALG-OAD). Soient $o \triangleq |\mathcal{U}^*|$ et $A \triangleq |\mathcal{U}|$. La preuve consiste à montrer que $o \leq 2A$.

Nous divisons les utilisateurs dans $\mathcal{U}^*$ en deux ensembles disjoints; $\mathcal{U}_1^*$ et $\mathcal{U}_2^*$. L'ensemble $\mathcal{U}_1^*$ représente les utilisateurs associés uniquement par OPT-OAD alors que l'ensemble $\mathcal{U}_2^*$ représente les utilisateurs associés par OPT-OAD et ALG-OAD. Soient $o_1 \triangleq |\mathcal{U}_1^*|$ et $o_2 \triangleq |\mathcal{U}_2^*|$. Par définition, nous avons $o_2 \leq A$. Puisque $o_1 + o_2 = o$, la preuve est réduite à montrer que $o_1 \leq A$. Pour prouver ceci, nous partitionnons $\mathcal{U}_1^*$ (resp. $\mathcal{U}$) en n ensembles disjoints $\mathcal{U}_1^*(b)$ (resp. $\mathcal{U}(b)$) pour tout $b = 1, \dots, n$. Ici, $\mathcal{U}_1^*(b)$ représente l'ensemble des utilisateurs associés à b uniquement par OPT-OAD alors que $\mathcal{U}(b)$ est l'ensemble des utilisateurs associés à b par ALG-OAD. Soient $o_1(b) \triangleq |\mathcal{U}_1^*(b)|$ et $A(b) \triangleq |\mathcal{U}(b)|$.

Proposition 4.5.5 $o_1(b) \leq A(b)$, pour tout $b = 1, \dots, n$.

Cette proposition, une fois établie, prouve que $O_1 \leq A$ en faisant la somme sur b des n inégalités précédentes, puisque $A = \sum_{b=1}^n A(b)$ et $O = \sum_{b=1}^n O_1(b)$. Il reste à la prouver. Soit b une station de base quelconque. Premièrement, pour tout utilisateur u^* dans $\mathcal{U}_1^*(b)$, il doit exister un utilisateur u dans $\mathcal{U}(b)$ tel que u interfère avec et termine plus tôt que u^* (sinon, ALG-OAD aurait associé u^* à la place de u). Deuxièmement, les utilisateurs de $\mathcal{U}_1^*(b)$ n'interfèrent pas entre eux. Donc, au plus un utilisateur de $\mathcal{U}_1^*(b)$ peut interférer avec un utilisateur de $\mathcal{U}(b)$. Par conséquent, $O_1(b) \leq A(b)$, pour tout $b = 1, \dots, n$. En conclusion, $O \leq 2A$. Ceci prouve le théorème. ■

Algorithmes heuristiques

Étant donné que OAD est lié au problème d'ordonnancement avec intervalles (Garey et Johnson, 1979), nous adaptons trois algorithmes heuristiques qui résolvent le problème d'ordonnancement avec intervalles afin de les comparer avec ALG-OAD. Le premier algorithme est EDF-OAD qui trie les utilisateurs en ordre croissant de dates limites et, en cas d'égalité, choisit ceux qui ont les plus petites dates d'arrivée. Le deuxième algorithme, appelé HR1-OAD, utilise la différence entre les dates limites et les dates d'arrivée pour trier les utilisateurs (c'est-à-dire, HR1-OAD trie les utilisateurs par ordre croissant de $d_u - r_u$). Enfin, le troisième algorithme, appelé HR2-OAD trie les utilisateurs en se basant sur la différence entre les dates limites et les dates d'arrivée plus le nombre de *slots* requis (c'est-à-dire, pour chaque station de base b , HR2-OAD trie les utilisateurs par ordre croissant de $d_u - r_u + v_{b \rightarrow u}$).

Contrairement à ALG-OAD, ces algorithmes heuristiques ne garantissent aucune performance dans le pire cas. Par exemple, pour EDF-OAD, supposons qu'il existe une seule station de base b et $k + 1$ utilisateurs avec des dates d'arrivée identiques, c.-à-d. $r_u = 0$ pour tout u . Supposons aussi que la date limite de l'utilisateur 1 est $d_1 = 4k$ et son nombre de *slots* requis est $v_{b \rightarrow 1} = 4k$ alors que les dates limites (resp. le nombre de *slots* requis) des utilisateurs restants sont fixées à $d_u = 4k + 2$ (resp. $v_{b \rightarrow u} = 4$) pour tout $u \in \{2, 3, \dots, k+1\}$.

Soient $m = (4k+2)$ slots. Dans cet exemple spécifique, EDF-OAD associe uniquement l'utilisateur 1 car il a la plus petite date limite et $m - v_{b \rightarrow 1} < v_{b \rightarrow u}$ pour tout $u > 1$. Cependant, OPT-OAD et ALG-OAD associent tous les k utilisateurs $u \in \{2, 3, \dots, k+1\}$. Nous pouvons voir que lorsque k augmente, la performance d'EDF-OAD diverge de l'optimal.

4.5.3 Résultats de simulation

Dans cette partie, nous présentons les résultats de simulation pour illustrer les performances des algorithmes proposés pour résoudre OAD. Nous considérons une zone géographique sous forme d'un carré de côté égal à 500 mètres. Nous considérons une macro-cellule et $n - 1$ petites cellules. Les stations de base et les utilisateurs sont générés aléatoirement d'une façon uniforme dans cette zone. Les gains des canaux sans fil sont basés sur les spécifications 3GPP (3GPP, 2017). Plus précisément, pour chaque utilisateur u , $g_{1 \rightarrow u} = 15,3 + 37,6 \cdot \log_{10} d_{1 \rightarrow u}$ pour la macro-cellule et $g_{b \rightarrow u} = 30,6 + 36,7 \cdot \log_{10} d_{b \rightarrow u}$ pour les petites cellules⁶. Nous considérons une fréquence de porteuse de 2 GHz et une bande passante $B = 10$ MHz. La puissance de transmission de la macro-cellule est 46 dBm, et les puissances de transmission des petites cellules sont $p_b = 30$ dBm pour tout b . La puissance du bruit AWGN est -174 dBm/Hz (Zhuang *et al.*, 2016). Le nombre de slots est $m = 10$ où chaque slot est de longueur $\tau = 0,1$ secondes. Chaque utilisateur u a besoin de télécharger $w_u = 1$ mégaoctet. Les simulations sont moyennées sur 1000 réalisations indépendantes. Dans chaque réalisation : (i) les positions des utilisateurs et des stations de base sont régénérées de manière aléatoire et uniforme; (ii) le date d'arrivée r_u est aléatoire selon une distribution de Poisson avec le paramètre $\lambda = 2$; et (iii) la date limite d_u est choisie égale à $r_u + X$ où X est une variable aléatoire uniforme qui appartient à $(0, d]$ (d représente une date limite commune qui est, sauf indication contraire, fixée à 10). Pour obtenir OPT-OAD, nous modélisons (4.5) en utilisant Python avec PuLP (Mit-

6. Nous utilisons $d_{b \rightarrow u}$ pour noter la distance en mètres entre b et u .

chell *et al.*, 2011) et nous utilisons GUROBI (Gurobi Optimization Inc., 2016) pour le résoudre.

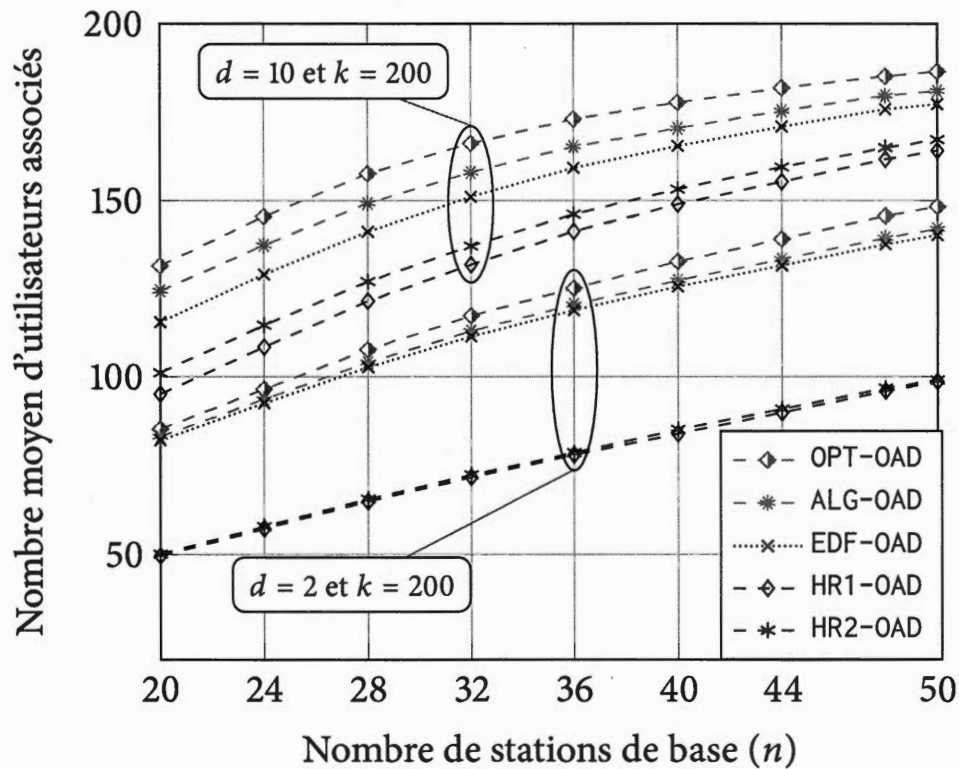


Figure 4.3 Nombre moyen d'utilisateurs associés par tous les algorithmes en fonction de n .

La figure 4.3 compare tous les algorithmes en fonction de n et d . Il est clair que la performance de ALG-OAD est proche de celle de OPT-OAD et que ALG-OAD surpasse tous les algorithmes heuristiques. Nous concluons que la performance de ALG-OAD est mieux que celle théorique qui prévoit un facteur d'approximation de 2. Nous pouvons voir que lorsque d augmente, plus d'utilisateurs sont associés puisque leurs dates limites sont plus grandes. Par conséquent, il existe un compromis entre la date limite commune d et le nombre d'utilisateurs associés.

Dans la figure 4.4, nous comparons ALG-OAD et OPT-OAD en variant k et n . L'écart

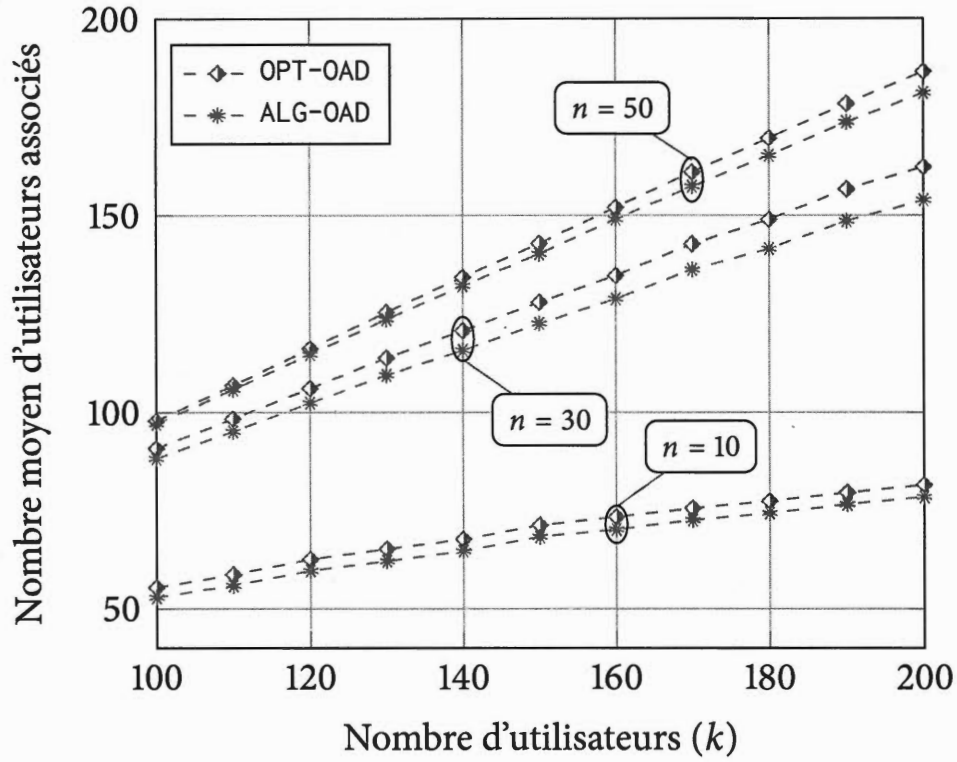


Figure 4.4 Nombre moyen d'utilisateurs associés par OPT-OAD et ALG-OAD en fonction de k .

de performance entre ALG-OAD et OPT-OAD est limité et reste pratiquement identique, même pour des grandes valeurs de k et n . Nous remarquons aussi que la performance de ALG-OAD est mieux que celle théorique qui fournit un facteur d'approximation de 2.

Pour mesurer l'équité entre les utilisateurs des algorithmes ALG-OAD et OPT-OAD, l'indice de Jain (Hobfeld *et al.*, 2017; Jain *et al.*, 1984) est utilisé. Étant donnée un vecteur $\mathbf{a} = [a_1, a_2, \dots, a_n]$, l'indice de Jain de $\mathbf{a}$ est donné par :

$$\frac{(\sum_{i=1}^n a_i)^2}{n \sum_{i=1}^n a_i^2}.$$

L'association la plus équitable est celle qui associe les utilisateurs le même nombre de

fois. La figure 4.5 montre que, lorsque k augmente, l'équité en utilisant ALG-OAD dimi-

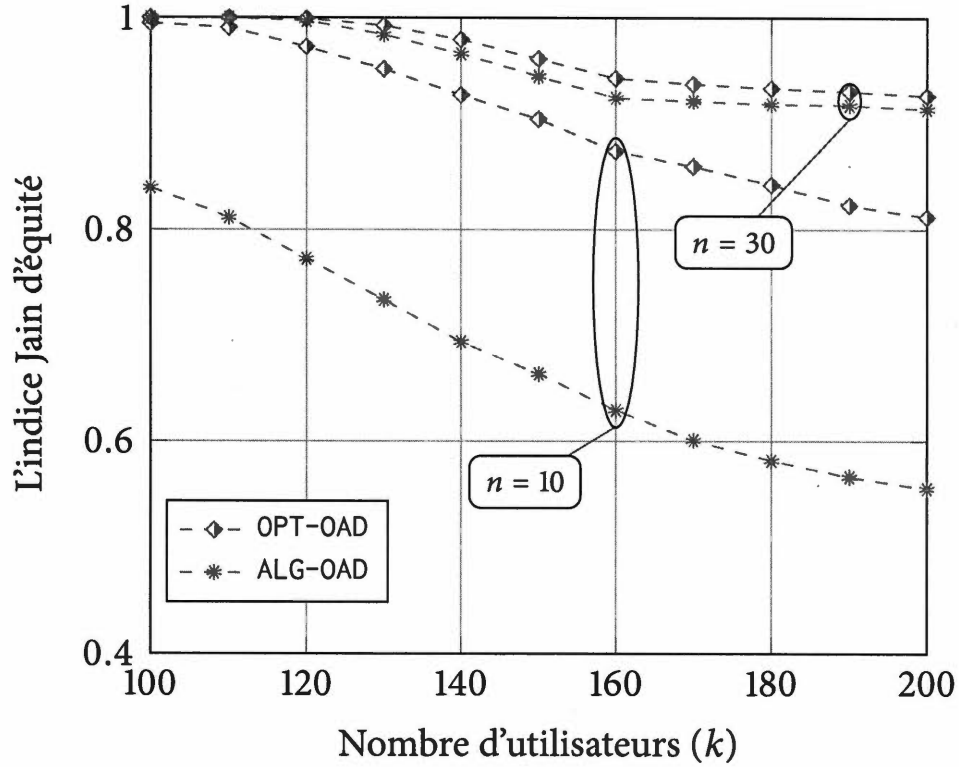


Figure 4.5 Équité d'utilisateurs par OPT-OAD et ALG-OAD en fonction de k .

nie pour les petites valeurs de n . Heureusement, lorsque n augmente, l'équité en utilisant ALG-OAD augmente puisque les utilisateurs seront fréquemment associés. Donc, ALG-OAD améliore l'équité dans les réseaux DCNs.

La figure 4.6 illustre l'efficacité spectrale moyenne (*average spectral efficiency (ASE)* en anglais) de ALG-OAD en la comparant à celle d'OPT-OAD et d'OPT-OAD_{SE} pour $n = 30$. OPT-OAD_{SE} est un algorithme optimal pour l'efficacité spectrale obtenu en modifiant OPT-OAD et en changeant la fonction objectif de (4.5) à $\sum_{u \in \mathcal{U}} \sum_{b \in \mathcal{B}} \sum_{s=r_u}^{d_u - \nu_{b \rightarrow u}} R_{b \rightarrow u} x_{b \rightarrow u}^s$. Même si ALG-OAD et OPT-OAD ne sont pas conçus pour optimiser l'efficacité spectrale, nous pouvons voir que leurs performances en termes d'efficacité spectrale sont

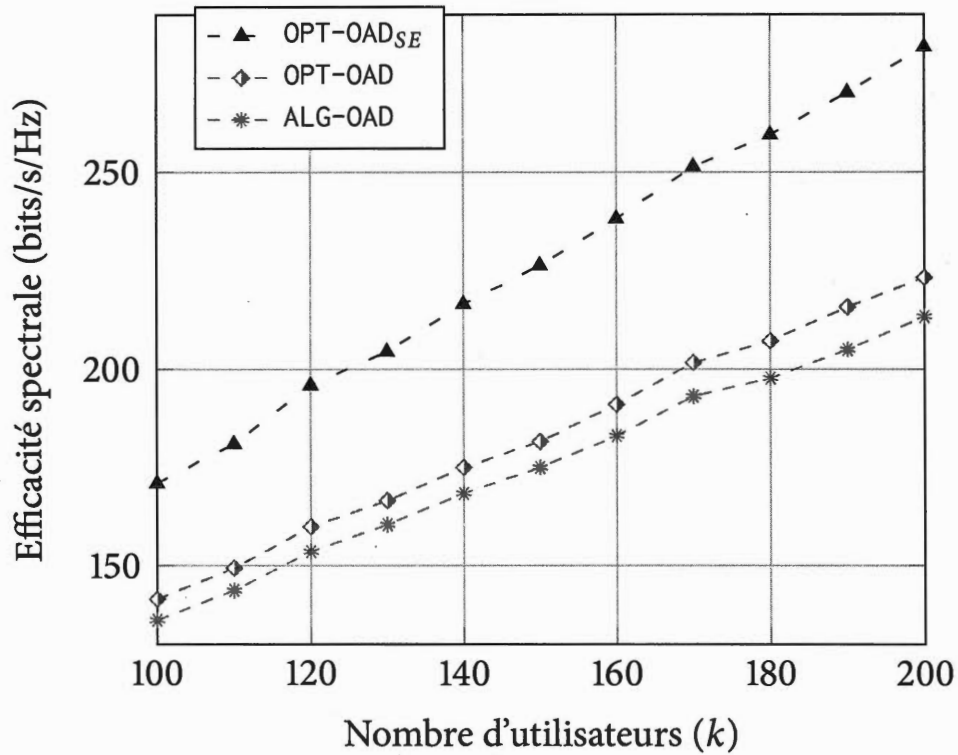


Figure 4.6 Efficacité spectrale par tous les algorithmes en fonction de k pour OAD.

très proches de celle de OPT-OAD_{SE}.

4.6 Le problème OAFED

Dans cette section, nous considérons PAR dans le cas d'ordonnement, d'association des utilisateurs et d'allocation de fréquences avec récolte d'énergie (OAFED). Contrairement à OAD, dans OAFED, la décision est prise au début de la trame où tous les utilisateurs sont supposés arriver durant la trame précédente (c'est comme si tous les utilisateurs ont les mêmes dates d'arrivées).

4.6.1 Formulation du problème

Dans cette section, nous étudions OAFED lorsque les stations de base utilisent la récolte d'énergie. Nous introduisons les variables binaires suivantes :

$$x_{b \rightarrow u}^{s,f} \triangleq \begin{cases} 1, & \text{si } u \text{ est associé à } b \text{ sur } f \text{ pendant } s \\ 0, & \text{sinon.} \end{cases}$$

Soit $z_{b,s}$ la variable entière qui représente le bilan accumulé de nombre de *slots* par la station de base b au *slot* s . Autrement dit, $0 \leq z_{b,s} \leq A_{b,s}$ et $z_{b,s} > 0$ si et seulement si b a une quantité d'énergie strictement positive disponible dans sa batterie.

Les contraintes de OAFED sont données dans la liste suivante :

- L'utilisateur u ne peut pas utiliser plus qu'une fréquence lorsqu'il est associé à b ,

$$x_{b \rightarrow u}^{s,f} + x_{b \rightarrow u}^{s',f'} \leq 1, \quad \forall f \neq f', (u, b, s, s', f, f') \in \mathcal{U} \times \mathcal{B} \times \mathcal{S}^2 \times \mathcal{F}^2. \quad (4.6)$$

- Les utilisateurs associés à b sur f ne peuvent pas être servis pendant le même *slot*,

$$\sum_{u \in \mathcal{U}} x_{b \rightarrow u}^{s,f} \leq 1, \quad \forall (b, s, f) \in \mathcal{B} \times \mathcal{S} \times \mathcal{F}. \quad (4.7)$$

- L'utilisateur u est associé à une seule station de base sur la fréquence f ,

$$x_{b \rightarrow u}^{s,f} + x_{b' \rightarrow u}^{s',f'} \leq 1, \quad \forall b \neq b', (u, b, b', s, s', f, f') \in \mathcal{U} \times \mathcal{B}^2 \times \mathcal{S}^2 \times \mathcal{F}^2. \quad (4.8)$$

- À chaque *slot* s , le nombre de *slots* cumulés est mis à jour,

$$z_{b,s+1} = z_{b,s} + A_{b,s+1} - \sum_{u \in \mathcal{U}} \sum_{f \in \mathcal{F}} x_{b \rightarrow u}^{s,f}, \quad \forall (b, s) \in \mathcal{B} \times \mathcal{S} \setminus \{m\}, \quad (4.9)$$

et les conditions initiales sont :

$$z_{b,1} = A_{b,1}, \quad \forall b \in \mathcal{B}. \quad (4.10)$$

- Chaque utilisateur u est associé à une station de base b sur la fréquence f seulement lorsque b a une quantité d'énergie positive disponible à s ,

$$x_{b \rightarrow u}^{s,f} \leq z_{b,s}, \quad \forall (u, b, s, f) \in \mathcal{U} \times \mathcal{B} \times \mathcal{S} \times \mathcal{F}, \quad (4.11)$$

- Chaque utilisateur u demande $v_{b \rightarrow u}^f$ lorsqu'il est associé à b sur la fréquence f ,

$$x_{b \rightarrow u}^{s,f} \sum_{t \in \mathcal{S}} x_{b \rightarrow u}^{t,f} = v_{b \rightarrow u}^f x_{b \rightarrow u}^{s,f}, \quad \forall (u, b, s, f) \in \mathcal{U} \times \mathcal{B} \times \mathcal{S} \times \mathcal{F}. \quad (4.12)$$

Les contraintes (4.12) ne sont pas linéaires mais elles peuvent être facilement modélisées dans les solveurs modernes tel que CPLEX (IBM, 2010). Néanmoins, elles peuvent être théoriquement linéarisées en utilisant la méthode bigM (Schrijver, 1986) comme nous avons fait dans le chapitre 3. Par conséquent, elles peuvent être réécrites comme :

$$\sum_{t \in \mathcal{S}} x_{b \rightarrow u}^{t,f} \geq v_{b \rightarrow u}^f x_{b \rightarrow u}^{s,f}, \quad \forall (u, b, s, f) \in \mathcal{U} \times \mathcal{B} \times \mathcal{S} \times \mathcal{F}. \quad (4.13)$$

$$\sum_{t \in \mathcal{S}} x_{b \rightarrow u}^{t,f} \leq v_{b \rightarrow u}^f x_{b \rightarrow u}^{s,f} + m(1 - x_{b \rightarrow u}^{s,f}), \quad \forall (u, b, s, f) \in \mathcal{U} \times \mathcal{B} \times \mathcal{S} \times \mathcal{F}. \quad (4.14)$$

- L'utilisateur u est servi avant sa date limite,

$$d_u \geq s x_{b \rightarrow u}^{s,f}, \quad \forall (u, b, s, f) \in \mathcal{U} \times \mathcal{B} \times \mathcal{S} \times \mathcal{F}. \quad (4.15)$$

OAFED a pour objectif de maximiser le nombre d'utilisateurs servis. Par conséquent,

la fonction objectif peut être écrite comme :

$$\sum_{u \in \mathcal{U}} \sum_{b \in \mathcal{B}} \sum_{s \in \mathcal{S}} \sum_{f \in \mathcal{F}} x_{b \rightarrow u}^{s,f} / v_{b \rightarrow u}^f. \quad (4.16)$$

La division par $v_{b \rightarrow u}^f$ dans (4.16) est en fait nécessaire puisque chaque utilisateur utilise $v_{b \rightarrow u}^f$ slot, s'il est associé.

Finalement, OAFED est formulé comme le programme linéaire en nombres entiers suivant :

$$\text{maximiser} \quad \sum_{u \in \mathcal{U}} \sum_{b \in \mathcal{B}} \sum_{s \in \mathcal{S}} \sum_{f \in \mathcal{F}} x_{b \rightarrow u}^{s,f} / v_{b \rightarrow u}^f \quad (4.17a)$$

$$\text{sous les contraintes} \quad x_{b \rightarrow u}^{s,f} \in \{0, 1\}, z_{b,s} \geq 0, \quad \forall (u, b, s, f) \in \mathcal{U} \times \mathcal{B} \times \mathcal{S} \times \mathcal{F}, \quad (4.17b)$$

(4.6) – (4.15).

En utilisant des solveurs comme CPLEX (IBM, 2010), nous pouvons obtenir une solution optimale à OAFED en résolvant (4.17). Dans ce qui suit, nous analysons la NP-difficulté de OAFED dans différents cas. Nous notons OAFED dans le cas X par OAFED-X. Un algorithme qui résout OAFED-X est noté par ALG-X et un algorithme optimal (obtenu en résolvant (4.17)) est noté par OPT-X.

4.6.2 NP-difficulté et solutions proposées

4.6.2.1 NP-difficulté

Afin de caractériser la NP-difficulté de OAFED, nous analysons quatre cas : (i) une Seule Fréquence et une Seule station de Base (SFSB), (ii) une Seule Fréquence et Multiples stations de Base (SFMB), (iii) Multiples Fréquences et une Seule station de Base (MFSB), et (iv) Multiples fréquences et Multiples stations de Base (MFMB). Nous résu-

mons les résultats de notre analyse dans le tableau 4.1 dans lequel nous présentons aussi les algorithmes proposés ainsi que leurs complexités asymptotiques. Nous définissons $q \triangleq \min\{k, n\}$.

Tableau 4.1 NP-difficulté, algorithmes et complexité asymptotique de OAFED

OAFED-SFSB	OAFED-SFMB	OAFED-MFSB	OAFED-MFMB
P	NP-difficile	NP-difficile	NP-difficile
ALG-SFSB	ALG-SFMB	ALG-MFSB	ALG-MFMB
Optimal	Approximation	Heuristique	Heuristique
$\mathcal{O}(km^2 + mk^2)$	$\mathcal{O}(nq(km^2 + mk^2))$	$\mathcal{O}(k^2m + k\ell m^2)$	$\mathcal{O}(nq(k^2m + k\ell m^2))$

En se basant sur la démonstration du théorème 4.5.3, nous pouvons conclure que OAFED-SFMB ainsi que OAFED-MFMB sont NP-difficiles. Il reste à étudier la NP-difficulté d'OAFED-SFSB et d'OAFED-MFSB. Nous commençons par OAFED-MFSB.

Théorème 4.6.1 *OAFED-MFSB est NP-difficile.*

Démonstration : soient $\mathcal{B} = \{1\}$, $\ell = 2$, $d_u = m$ pour tout u , $A_{1,s} = m$ pour tout s . Autrement dit, il existe une station de base $b = 1$ qui a assez d'énergie pendant tous les *slots*, un ensemble d'utilisateurs $\{1, 2, \dots, k\}$ et deux fréquences. Nous réduisons en temps polynomial PARTITION (Garey et Johnson, 1979) à une version de décision de OAFED-MFSB. Dans PARTITION, il existe un ensemble de r entiers positifs $\mathcal{N} = \{a_1, \dots, a_r\}$ que nous devons partitionner en deux sous-ensembles disjoints $\mathcal{N}_1$ et $\mathcal{N}_2$ tels que $\mathcal{N}_1 \cup \mathcal{N}_2 = \mathcal{N}$ et $\sum_{i \in \mathcal{N}_1} a_i = \sum_{i \in \mathcal{N}_2} a_i$. Étant donnée une instance de PARTITION, nous construisons une instance de la version de décision de OAFED-MFSB comme suit. Soit $m = \lfloor \sum_{i \in \mathcal{N}} a_i / 2 \rfloor$ (supposons sans perte de généralité que $m = \sum_{i \in \mathcal{N}} a_i / 2$), $k = r$ et $v_{1 \rightarrow u}^1 = v_{1 \rightarrow u}^2 = a_u$ pour tout u . La question qui se pose dans la version de décision de OAFED-MFSB est la suivante : étant donnée cette instance, pouvons-nous servir tous les utilisateurs ?

D'une part, si nous pouvons résoudre PARTITION, alors nous pouvons choisir les éléments de $\mathcal{N}_1$ (resp. $\mathcal{N}_2$) comme les utilisateurs servis à partir du premier *slot* sur la fréquence 1 (resp. la fréquence 2). Ainsi, la version de décision de OAFED-MCSB est résolue. D'autre part, si nous pouvons servir tous les utilisateurs sur les deux fréquences, alors les utilisateurs servis sur la fréquence 1 (resp. la fréquence 2) peuvent être choisis pour représenter les éléments de $\mathcal{N}_1$ (resp. $\mathcal{N}_2$). Il est clair que $\mathcal{N}_1 \cup \mathcal{N}_2 = \mathcal{N}$. De plus, puisque tous les utilisateurs sont servis et que $m = \sum_{i \in \mathcal{N}} a_i / 2$, alors les utilisateurs servis sur la fréquence 1 (ou la fréquence 2) demandent exactement $\sum_{i \in \mathcal{N}} a_i / 2$. Par conséquent, $\sum_{i \in \mathcal{N}_1} a_i = \sum_{i \in \mathcal{N}_2} a_i = \sum_{i \in \mathcal{N}} a_i / 2$. Ainsi, PARTITION est résolu.

Nous pouvons voir que l'instance de la version de décision de OAFED-MFSB est créée en temps polynomial. Puisque PARTITION est NP-difficile, alors OAFED-MFSB l'est aussi. Cela prouve le théorème. ■

Quant à OAFED-SFSB, nous montrons dans ce qui suit qu'il appartient à la classe de complexité P.

4.6.2.2 Solutions proposées

OAFED-SFSB

Cette partie considère OAFED-SFSB. L'étude d'OAFED-SFSB est importante, car cela nous permet de caractériser la structure des solutions optimales dans des cas plus généraux. Nous présentons sur la figure 4.7 un exemple d'une solution faisable à OAFED-SFSB.

Pour résoudre OAFED-SFSB, nous proposons un algorithme polynomial optimal, appelé ALG-SFSB, qui ordonnance un nombre maximum d'utilisateurs tout en respectant leurs dates limites ainsi que les contraintes d'énergie. Tout d'abord, nous décrivons ALG-SFSB, puis nous prouvons son optimalité.

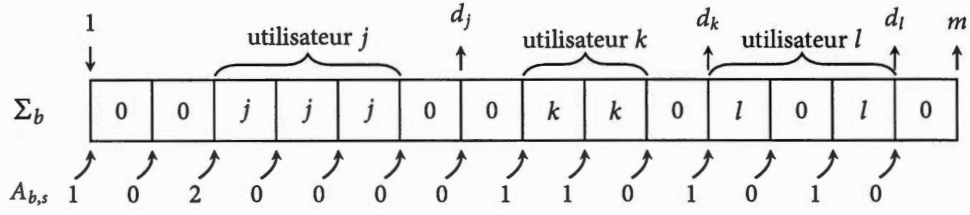


Figure 4.7 Un exemple d'un ordonnanceur Σ_b d'une station de base b .

Nous commençons par les définitions suivantes :

Définition 4.6.2 (Un ordonnanceur faisable) :

Un ordonnanceur est dit faisable si tous les utilisateurs ordonnancés respectent leurs dates limites, leurs demandes ainsi que les contraintes d'énergie.

Définition 4.6.3 (Un ordonnanceur ϵ -optimal) :

Un ordonnanceur $\Sigma^\epsilon = [\sigma_1^\epsilon, \sigma_2^\epsilon, \dots, \sigma_m^\epsilon]$ est dit ϵ -optimal, s'il s'agit d'un ordonnanceur faisable pour les utilisateurs $\{1, 2, \dots, \epsilon\}$ et il ordonnance le nombre maximum d'utilisateurs parmi $\{1, 2, \dots, \epsilon\}$.

ALG-SFSB est décrit dans le pseudo-code présenté dans l'algorithme 8. Il fonctionne comme suit. Premièrement, il trie les utilisateurs comme dans EDF. Ensuite, il commence par un ordonnanceur inoccupé Σ (c'est-à-dire, $\Sigma = [0, 0, \dots, 0]$) et itère l'ensemble des utilisateurs triés (qui sont numérotés une fois triés) tout en vérifiant les contraintes d'énergie et les dates limites. Soit $\mathcal{S}^{u-1}$ l'ensemble d'utilisateurs déjà ordonnancés par Σ avant le début de la u -ème itération. À la u -ème itération, l'utilisateur u est celui qui va être ordonnancé et donc ALG-SFSB l'ajoute à $\mathcal{S}^{u-1}$, c.-à-d. ALG-SFSB crée l'ensemble $\mathcal{S}^u = \mathcal{S}^{u-1} \cup \{u\}$. Ensuite, ALG-SFSB itère les *slots* $s = 1, 2, \dots, m$ et ordonnance u à un *slot* s' chaque fois que (i) $A_s > 0$ pour $s \leq s'$ et (ii) $\sigma_{s'} = 0$. Si $A_s = 0$, alors le prochain *slot* est considéré. Après l'ordonnancement de l'utilisateur u , ALG-SFSB vérifie trois conditions pour voir si les contraintes d'énergie, de dates limites et des demandes des utilisateurs sont respectées ou pas. Ces conditions sont données dans la liste suivante.

Algorithme 8 L'algorithme ALG-SFSB

```

1: fonction SFSB( $\mathcal{U}, \mathbf{d}, \mathbf{A}, \nu$ )
2:   Trier les utilisateurs par ordre croissant des dates limites
3:   Soit  $\Sigma$  ordonnanceur inoccupé, c.-à-d.,  $\Sigma \leftarrow [0, \dots, 0]$ 
4:    $u \leftarrow 1$ 
5:    $\mathcal{S}^0 \leftarrow \emptyset$ 
6:   pour  $u \leftarrow 1$  à  $k$  faire
7:      $\mathcal{S}^u \leftarrow \mathcal{S}^{u-1} \cup \{u\}$ 
8:      $x \leftarrow 0$ 
9:      $s \leftarrow 1$ 
10:     $r \leftarrow 1$ 
11:    tant que  $s \leq m$  faire
12:       $r \leftarrow \max\{r, s\}$ 
13:      si  $A_s > 0$  alors
14:         $\delta \leftarrow \min\{A_s, \nu_u - x, m - r + 1\}$ 
15:        pour  $s'$  dans  $\{r, \dots, r + \delta - 1\}$  faire
16:          si  $\sigma_{s'} = 0$  alors
17:             $\sigma_{s'} \leftarrow u$ 
18:             $x \leftarrow x + 1$ 
19:             $r \leftarrow r + 1$ 
20:             $A_s \leftarrow A_s - 1$ 
21:          fin si
22:        fin pour
23:      sinon
24:         $s \leftarrow s + 1$ 
25:      fin si
26:      si  $\nu_u = x$  et  $d_u \geq r - 1$  alors
27:        sortir
28:      sinon si  $\nu_u = x$  et  $d_u < r - 1$  alors
29:         $l \leftarrow \arg \max\{\nu_i : i \in \mathcal{S}^u\}$ 
30:         $(\Sigma, \mathbf{A}, \mathcal{S}^u, s, r) \leftarrow \text{UPDATE}(\Sigma, \mathbf{A}, \mathcal{S}^u, l)$ 
31:        sortir
32:      sinon si  $\nu_u > x$  et  $\max\{r, s\} > m$  alors
33:         $l \leftarrow \arg \max\{\nu_i : i \in \mathcal{S}^u\}$ 
34:         $(\Sigma, \mathbf{A}, \mathcal{S}^u, s, r) \leftarrow \text{UPDATE}(\Sigma, \mathbf{A}, \mathcal{S}^u, l)$ 
35:        si  $l = u$  alors
36:          sortir
37:        fin si
38:      fin si
39:    fin tant que
40:  fin pour
41:  retourner  $(\Sigma, \mathcal{S}^k)$ 
42: fin fonction

```

1. À l'utilisateur u sont alloués v_u *slots* et sa date limite d_u est respectée. Dans ce cas, ALG-SFSB passe à l'utilisateur $u + 1$.
2. À l'utilisateur u sont alloués v_u *slots* mais sa date limite d_u n'est pas respectée. Dans ce cas, ALG-SFSB enlève de S^u l'utilisateur l qui demande le plus grand nombre de *slots*, il met à jour Σ et A et passe à l'utilisateur $u + 1$.
3. À l'utilisateur u on n'a pas encore alloué v_u *slots* (car il n'y a pas assez d'énergie) et il ne reste plus de temps dans la trame. Dans ce cas, ALG-SFSB enlève de S^u l'utilisateur l qui demande le plus grand nombre de *slots*, il met à jour Σ et A et passe à l'utilisateur $u + 1$ uniquement si $l = u$. En d'autres termes, ALG-SFSB passe à l'utilisateur $u + 1$ s'il a enlevé l'utilisateur u , mais continue dans l'ordonnancement de u sinon.

Dans les deux dernières conditions, une procédure de re-ordonnancement, appelée UPDATE, est exécutée. Elle effectue principalement trois opérations : (1) enlève l de S^u , (2) décale tous les utilisateurs planifiés après l vers la gauche et (3) met à jour l'énergie (ou A). Le pseudo-code présenté dans l'algorithme 9 résume cette procédure de re-ordonnancement. Les deux dernières opérations de cette procédure sont décrites ci-après.

D'une part, la deuxième opération trouve principalement les ensembles de *slots* $S_1(l)$ et $S_2(l)$, où $S_1(l)$ représente les *slots* pendant lesquels l est ordonnancé alors que $S_2(l)$ est l'ensemble de *slots* pendant lesquels tous les utilisateurs $j \neq l$ sont ordonnancés après l , c'est-à-dire $S_1(l) = \{s \in \mathcal{S} : \sigma_s = l\}$ et $S_2(l) = \{s \in \mathcal{S} : \sigma_s \neq 0 \text{ et } \sigma_s \neq l \text{ et } s \geq \min S_1\}$. Ensuite, pour chaque *slot* $s \in S_2(l)$, σ_s est décalé vers la gauche ou bien à un *slot* de $S_1(l)$ ou à un *slot* de $S_2(l)$ d'un utilisateur déjà décalé. Bien sûr, chaque fois qu'un utilisateur u est déplacé, UPDATE doit s'assurer qu'il est ordonnancé à un *slot* inoccupé et qu'il y a suffisamment d'énergie disponible (sa date limite sera respectée puisqu'il sera décalé vers la gauche). Ce décalage d'utilisateurs crée un nouveau ordonnanceur Σ dans lequel il existe des nouveaux *slots* inoccupés. D'autre part, la troisième opération calcule l'ensemble des

nouveaux *slots* inoccupés $\mathcal{S}_3(l)$. Pour chaque *slot* $s \in \mathcal{S}_3(l)$, il y avait un utilisateur qui a été ordonnancé à s et utilisait une unité d'énergie à un *slot* précédent $s' \leq s$ (avant le décalage). Ainsi, pour chaque *slot* $s \in \mathcal{S}_3(l)$, la troisième opération trouve le *slot* s' et met à jour l'énergie comme $A_{s'} \leftarrow A_{s'} + 1$.

Après avoir effectué ces opérations, ALG-SFSB retourne le tuple $(\Sigma, A, \mathcal{S}^u, s, r)$.

Algorithme 9 La procédure de re-ordonnancement

```

1: fonction UPDATE( $\Sigma, A, \mathcal{S}^u, l$ )
2:    $\mathcal{S}^u \leftarrow \mathcal{S}^u \setminus \{l\}$ 
3:   Mettre à jour  $\Sigma$  en décalant vers la gauche les utilisateurs ordonnancés après  $l$ 
4:   Mettre à jour l'énergie  $A$  // La mise à jour de  $A$  : (i) diminue l'énergie
   nécessaire pour ordonnancer un utilisateur, ou (ii) rajoute l'énergie utilisée pour
   ordonnancer un ancien utilisateur qui a été supprimé.
5:    $s \leftarrow 1$ 
6:    $r \leftarrow 1$ 
7:   retourner  $(\Sigma, A, \mathcal{S}^u, s, r)$ 
8: fin fonction

```

Dans ce qui suit, nous prouvons que ALG-SFSB retourne toujours un ordonnanceur optimal. Tout d'abord, nous montrons que cet ordonnanceur est faisable, puis nous prouvons son optimalité. Soit Σ^u l'ordonnanceur retourné par ALG-SFSB après l'itération u , c'est-à-dire, Σ^u correspond à l'ensemble $\mathcal{S}^u$ d'utilisateurs ordonnancés par ALG-SFSB. Tout au long de cette partie, nous appliquons la terminologie de la théorie des ensembles à Σ^u , par exemple, $\Sigma^{u-1} \cup \{u\}$ correspond à l'ensemble $\mathcal{S}^{u-1} \cup \{u\}$ d'utilisateurs ordonnancés par ALG-SFSB.

Lemme 4.6.4 *L'ordonnanceur Σ^u est faisable.*

Démonstration : nous prouvons le lemme par récurrence sur les utilisateurs. Pour $u = 1$, il est clair que Σ^1 est faisable, puisque Σ^1 contient ou bien l'utilisateur 1 ou il est vide. Dans le cas où il contient l'utilisateur 1, il est clair que ce dernier respecte sa date limite ainsi que les contraintes énergétiques. Supposons que Σ^{u-1} est faisable. Si $\Sigma^u = \Sigma^{u-1} \cup$

$\{u\}$, (autrement dit, la procédure de re-ordonnancement n'est pas exécutée), alors Σ^u est faisable et la preuve est terminée. Sinon (la procédure de re-ordonnancement est exécutée), supposons que l représente l'utilisateur à enlever de $\Sigma^{u-1} \cup \{u\}$, c'est-à-dire que l est l'utilisateur qui demande le plus grand nombre de *slots*, appelé ci-après *le plus grand utilisateur*. Nous prouvons que $\Sigma^u = \Sigma^{u-1} \cup \{u\} \setminus \{l\}$ est un ordonnanceur faisable. Nous divisons les utilisateurs de Σ^u en trois ensembles disjoints : (i) ceux ordonnancés avant l , (ii) ceux ordonnancés après l et (iii) u . Pour chaque ensemble, nous avons :

- (i) les utilisateurs ordonnancés avant l restent inchangés (p. ex. l'utilisateur j dans la figure 4.8);
- (ii) les utilisateurs ordonnancés après l terminent plus tôt en raison de l'opération de décalage de la procédure de re-ordonnancement. Par conséquent, ces utilisateurs respectent leurs dates limites. De plus, ils respectent les contraintes énergétiques puisque l est le plus grand utilisateur et que Σ^{u-1} est faisable (p. ex. les utilisateurs i et u de la figure 4.8); et
- (iii) l'utilisateur u respecte les contraintes d'énergie car $v_u \leq v_l$ et Σ^{u-1} est faisable. Aussi, u respecte sa date limite car (i) le dernier utilisateur de Σ^u termine avant le dernier utilisateur (disons i) de Σ^{u-1} en raison de l'opération de décalage et du fait que l est le plus grand utilisateur, (ii) $d_u \geq d_i$ puisque les utilisateurs sont triés comme dans EDF et (iii) Σ^{u-1} est faisable. La figure 4.8 donne une illustration de cette démonstration.

Donc, Σ^u est un ordonnanceur qui respecte les contraintes énergétiques, les demandes des utilisateurs ainsi que les dates limites des utilisateurs. Donc, il s'agit d'un ordonnanceur faisable. Ceci prouve le lemme. ■

Théorème 4.6.5 *L'ordonnanceur Σ^u est u -optimal pour tout $u = 1, 2, \dots, k$.*

Démonstration : nous prouvons d'abord que pour $l > u$, il existe un ordonnanceur Σ'' qui est l -optimal composé des utilisateurs de Σ^u et des utilisateurs de $\{u+1, u+2, \dots, l\}$.

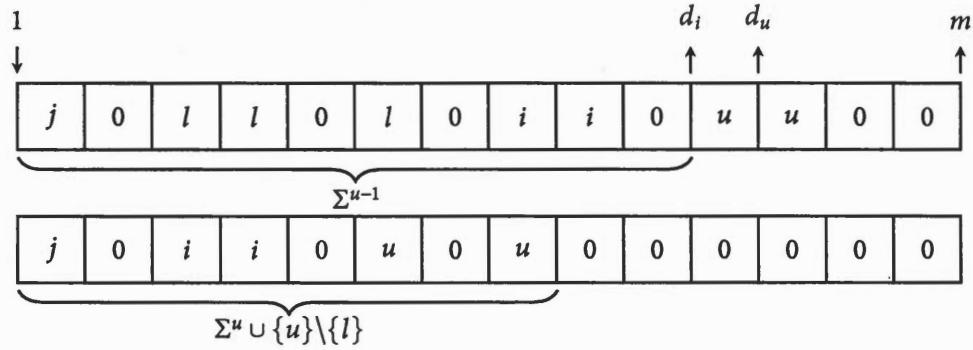


Figure 4.8 Une illustration de la démonstration du lemme 4.6.4.

Nous procédons par récurrence. Supposons qu'il existe un ordonnanceur Σ' qui est l -optimal composé des utilisateurs de Σ^{u-1} et des utilisateurs de $\{u, u+1, \dots, l\}$. Nous avons trois cas :

- (i) Si $\Sigma^u = \Sigma^{u-1} \cup \{u\}$. Dans ce cas, c'est clair que l'ordonnanceur $\Sigma'' = \Sigma'$ est l -optimal et il est composé des utilisateurs de Σ^u et des utilisateurs de $\{u+1, u+2, \dots, l\}$.
- (ii) Si $\Sigma^u = \Sigma^{u-1} \cup \{u\} \setminus \{j\}$ où $j \notin \Sigma'$. Dans ce cas, c'est clair aussi que l'ordonnanceur $\Sigma'' = \Sigma'$ est l -optimal et il est composé des utilisateurs de Σ^u et des utilisateurs de $\{u+1, u+2, \dots, l\}$.
- (iii) Si $\Sigma^u = \Sigma^{u-1} \cup \{u\} \setminus \{j\}$ où $j \in \Sigma'$. Dans ce cas, pour obtenir Σ'' , nous devons modifier Σ' . Par hypothèse, $\Sigma^{u-1} \cup \{u\}$ est non-faisable. Donc, il existe $i \in \Sigma^{u-1} \cup \{u\}$ tel que $i \notin \Sigma'$. Soit $\Sigma'' = \Sigma' \cup \{i\} \setminus \{j\}$. C'est clair que Σ'' est composé des utilisateurs de Σ^u et des utilisateurs de $\{u+1, u+2, \dots, l\}$. En outre, le nombre d'utilisateurs dans Σ' est égal à celui dans Σ'' . Ainsi, il reste à montrer que Σ'' est faisable. Nous notons que, puisque $i \in \Sigma^{u-1} \cup \{u\}$, alors Σ'' ne diffère de Σ' que dans son intersection avec les utilisateurs $\{1, 2, \dots, u\}$. Ceci montre que $\Sigma'' \cap \{1, 2, \dots, u\} \subseteq \Sigma^u$ et donc, en utilisant le lemme 4.6.4, Σ^u est faisable. Ainsi, $\Sigma'' \cap \{1, 2, \dots, u\}$ est faisable. En se basant sur l'argument précédent et sur le fait que ALG-SFSB enlève toujours le plus grand utilisateur (c.-à-d., $v_j \geq v_i$), nous concluons que Σ'' est fai-

sable. Finalement, nous avons : pour $l > u$, il existe un ordonnanceur l -optimal Σ'' composé des utilisateurs de Σ^u et des utilisateurs de $\{u + 1, u + 2, \dots, l\}$.

Nous utilisons le résultat précédent et la récurrence sur les utilisateurs pour prouver que Σ^u est u -optimal pour tout $u = 1, 2, \dots, k$. Pour $u = 1$, il est clairement vrai que Σ^1 est 1-optimal. Supposons que Σ^{u-1} est $(u - 1)$ -optimal. En appliquant le résultat précédent pour $l = u > u - 1$, nous savons qu'il existe un ordonnanceur u -optimal composé des utilisateurs de Σ^{u-1} et de $\{u\}$. Si $\Sigma^u = \Sigma^{u-1} \cup \{u\}$, alors Σ^u est u -optimal puisque, par hypothèse, Σ^{u-1} est $(u - 1)$ -optimal. Sinon, l'ordonnanceur u -optimal est (i) un sous-ensemble des utilisateurs de $\Sigma^{u-1} \cup \{u\}$ et (ii) a au moins autant d'utilisateurs que Σ^{u-1} parce que ce dernier est $(u - 1)$ -optimal. Clairement, $\Sigma^u \cup \{u\} \setminus \{j\}$, où j est le plus grand utilisateur, satisfait les deux conditions (i) et (ii) et est donc u -optimal. Par conséquent, Σ^u est u -optimal. Cela prouve le théorème. ■

Nous avons montré que ALG-SFSB retourne une solution optimale pour OAFED-SFSB. La complexité asymptotique de ALG-SFSB est $\mathcal{O}(km^2 + k^2m + k \log k) = \mathcal{O}(km^2 + k^2m)$ et elle est calculée comme suit. La boucle **pour** de la ligne 6 s'arrête dans $\mathcal{O}(k)$ itérations. Nous montrons que la boucle **tant que** de la ligne 11 s'arrête dans $\mathcal{O}(m)$ itérations. Soit s une itération arbitraire de cette boucle. Nous pouvons voir que si $A_s = 0$, alors le temps est incrémenté. Par conséquent, sans perte de généralité, supposons que $A_s > 0$ chaque fois que ALG-SFSB entre dans cette boucle. Cela dit, nous avons trois cas : (i) $v_u = x$ et $d_u \geq r - 1$, (ii) $v_u = x$ et $d_u < r - 1$ ou (iii) $v_u > x$ et $\max\{r, s\} > m$. Dans les deux premiers cas, la boucle **tant que** s'arrête en appelant **sortir** respectivement dans les lignes 27 et 31. Le dernier cas est le plus coûteux en termes de complexité temporelle. Chaque fois que $v_u > x$ et $\max\{r, s\} > m$, ALG-SFSB appelle la procédure de re-ordonnancement et passe à la prochaine itération à partir de $s = 1$. Mais nous pouvons voir que chaque fois que $A_s > 0$, ALG-SFSB incrémente x par au plus δ . Ainsi, à une certaine itération, la variable x sera supérieure ou égale à v_u . À ce stade, ALG-SFSB passe à l'une des deux conditions **si** précédentes de la boucle **tant que** s'arrête éventuellement.

Dans le pire cas, x est incrémenté par 1 à chaque fois, c'est-à-dire que $\delta = 1$. Par conséquent, la boucle **tant que** de la ligne 11 s'arrête dans $\mathcal{O}(m)$ itérations. Chaque itération de cette boucle a une complexité de $\mathcal{O}(k + m) - \mathcal{O}(k)$ pour trouver le plus grand utilisateur et $\mathcal{O}(m)$ pour exécuter la procédure de re-ordonnancement. Enfin, la complexité de ALG-SFSB dans le pire cas est donnée par $\mathcal{O}(km(k + m) + k \log k)$. Par conséquent, ALG-SFSB s'arrête dans $\mathcal{O}(km^2 + k^2m)$ itérations dans le pire cas.

OAFED-SFMB

Puisque OAFED-SFMB est NP-difficile, nous concevons un algorithme d'approximation afin de le résoudre efficacement. Nous proposons ALG-SFMB—un algorithme itératif qui fait appel à ALG-SFSB. Pour chaque itération, ALG-SFMB trouve la station de base qui est capable d'ordonnancer le nombre maximum d'utilisateurs. Le pseudo-code de ALG-SFMB est donné dans l'algorithme 10. Nous utilisons les notations supplémentaires suivantes. L'ensemble $\mathcal{A}_{\mathcal{B}}$ dénote $\{\mathcal{A}_i : i \in \mathcal{B}\}$, où $\mathcal{A}_i$ est un ensemble. De plus, étant donnée une matrice $\mathbf{A}$ (la matrice des *slots* récoltés $A_{b,s}$), nous utilisons $\mathbf{A}_i$ pour désigner sa i -ème ligne et $\mathbf{A}^i$ pour désigner sa i -ème colonne.

De manière plus détaillée, ALG-SFMB fonctionne comme suit. Soit $\mathcal{X}$ (resp. $\mathcal{Y}$) l'ensemble des utilisateurs restants (resp. l'ensemble des stations de base restantes) après chaque itération. Au départ, $\mathcal{X} = \mathcal{U}$ et $\mathcal{Y} = \mathcal{B}$. Pour chaque itération, ALG-SFMB itère $\mathcal{Y}$ et ordonnance les utilisateurs de $\mathcal{X}$ en utilisant ALG-SFSB. À la fin de cette itération, ALG-SFMB crée un ensemble $\mathcal{A}_b$ d'utilisateurs ordonnancés par chaque station de base b . Ensuite, il sélectionne b^* qui ordonnance le nombre maximum d'utilisateurs. Après, il met à jour l'ensemble $\mathcal{X}$ en supprimant les utilisateurs de $\mathcal{A}_{b^*}$ et met à jour l'ensemble $\mathcal{Y}$ en supprimant b^* . À chaque itération, ALG-SFMB crée l'ensemble $\mathcal{Z}$ des stations de base utilisées. À la fin, ALG-SFMB retourne l'ensemble $\mathcal{A}_{\mathcal{Z}}$ des utilisateurs servis.

Le théorème suivant montre qu'ALG-SFMB est un algorithme 2-approximation pour

Algorithme 10 L'algorithme ALG-SFMB

```

1: fonction SFMB( $\mathcal{U}, \mathcal{B}, \nu, \mathbf{A}$ )
2:    $\mathcal{X} \leftarrow \mathcal{U}$ 
3:    $\mathcal{Y} \leftarrow \mathcal{B}$ 
4:    $\mathcal{Z} \leftarrow \emptyset$ 
5:    $\mathcal{A} \leftarrow \emptyset$ 
6:   tant que  $\mathcal{X} \neq \emptyset$  et  $\mathcal{Y} \neq \emptyset$  faire
7:     pour  $b$  dans  $\mathcal{Y}$  faire
8:       Soit  $(\Sigma, \mathcal{A}_b) \leftarrow \text{SFSB}(\mathcal{X}, \mathbf{d}, \mathbf{A}_b, \nu_b)$ 
9:     fin pour
10:     $b^* \leftarrow \arg \max \{\mathcal{A}_y\}$ 
11:     $\mathcal{Y} \leftarrow \mathcal{Y} \setminus \{b^*\}$ 
12:     $\mathcal{X} \leftarrow \mathcal{X} \setminus \mathcal{A}_{b^*}$ 
13:     $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{b^*\}$ 
14:  fin tant que
15:  retourner  $\mathcal{A}_{\mathcal{Z}}$ 
16: fin fonction

```

OAFED-SFMB.

Théorème 4.6.6 ALG-SFMB est un algorithme 2-approximation pour OAFED-SFMB.

Démonstration : d'abord, ALG-SFMB est un algorithme polynomial puisque ALG-SFSB l'est et que la boucle **tant que** au niveau de la ligne 6 s'arrête en $q \triangleq \min(k, n)$ itérations. Ceci implique que la complexité asymptotique de ALG-SFMB est $O(nq(km^2 + k^2m))$ itérations dans le pire cas. Soit OPT-SFMB un algorithme optimal pour OAFED-SFMB. Soit $\mathcal{O}_b^1$ l'ensemble des utilisateurs associés à b uniquement par OPT-SFMB et notons par $o_b^1 = |\mathcal{O}_b^1|$. Soit $\mathbf{A}$ (resp. $\mathbf{A}_b$) le nombre d'utilisateurs associés à toutes les stations de base (resp. à la station de base b) par ALG-SFMB, c.-à-d. $\mathbf{A} \triangleq \sum_{b=1}^n |\mathcal{A}_b|$ (resp. $\mathbf{A}_b \triangleq |\mathcal{A}_b|$). Enfin, soit $\mathcal{O}_b^2$ l'ensemble des utilisateurs associés à la station de base b par OPT-SFMB et ALG-SFMB et notons $o_b^2 = |\mathcal{O}_b^2|$. Il est clair que $o \triangleq \sum_{b=1}^n o_b^1 + o_b^2$. Pour tout b , les utilisateurs de $\mathcal{O}_b^1$ peuvent être considérés par ALG-SFMB. Puisque ALG-SFMB associe le nombre maximum d'utilisateurs restants, alors b doit associer au moins o_b^1 , c'est-à-dire, $\mathbf{A}_b \geq o_b^1$, pour tout

$b = 1, 2, \dots, n$. Ainsi, en faisant la somme sur b , nous aurons :

$$A \geq \sum_{b=1}^n o_b^1. \quad (4.18)$$

Maintenant, il y a deux cas : le nombre total d'utilisateurs associés uniquement par OPT-SFMB est supérieur ou égal à la moitié du nombre optimal o . Autrement dit,

- $\sum_{b=1}^n o_b^1 \geq o/2$. Dans ce cas, en utilisant (4.18), nous obtenons $o \leq 2A$.
- $\sum_{b=1}^n o_b^1 \leq o/2$. Dans ce cas, nous avons $\sum_{b=1}^n o_b^1 + \sum_{b=1}^n o_b^2 \leq o/2 + \sum_{b=1}^n o_b^2$ et donc $o \leq o/2 + \sum_{b=1}^n o_b^2$. Autrement dit, $o/2 \leq \sum_{b=1}^n o_b^2$. Puisque les utilisateurs dans $\mathcal{O}_b^2$ sont associés par ALG-SFMB et OPT-SFMB, ainsi $o_b^2 \leq A_b$ pour tout b . Donc, $\sum_{b=1}^n o_b^2 \leq A$. Finalement, nous obtenons $o \leq 2A$.

Dans les deux cas, nous avons $A \leq o \leq 2A$. Par conséquent, ALG-SFMB est un algorithme 2-approximation pour OAFED-SFMB. Cela prouve le théorème. ■

Observation 4.6.7 *Le facteur d'approximation de 2 est le meilleur possible pour ALG-SFMB. Autrement dit, l'analyse faite dans le théorème 4.6.6 est étroite. Nous pouvons montrer ceci comme suit. Soit $k = 4$, $n = 2$, $m = 2$ et l'arrivée des slots est donnée par la matrice suivante :*

$$A = \begin{pmatrix} 2 & 0 \\ 2 & 0 \end{pmatrix}.$$

Les slots requis par les utilisateurs sont donnés par la matrice suivante :

$$v = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 4 \end{pmatrix}.$$

D'une part, OPT-SFMB associe les utilisateurs 1 et 4 à la station de base 1 et les ordonnance pendant les slots 1 et 2, respectivement, et associe les utilisateurs 2 et 3 à la station de base 2 et les ordonnance pendant les slots 1 et 2, respectivement. Par conséquent, $o = 4$. D'autre

part, ALG-SFMB peut associer les utilisateurs 2 et 3 à la station de base 1 et les ordonnance pendant les slots 1 et 2, respectivement, et donc il ne peut associer que l'utilisateur 1 (ou l'utilisateur 4) à la station de base 2 et l'ordonnance pendant le slot 1 (ou le slot 2). Par conséquent, $A = 2$. Nous pouvons voir que $O = 2A$.

On peut généraliser l'instance créée dans l'observation précédente pour montrer que le facteur d'approximation 2 de l'algorithme ALG-SFMB est serré.

Dans ce qui suit, nous considérons OAFED dans le cas de plusieurs fréquences (c'est-à-dire, OAFED-MFSB et OAFED-MFMB). Pour résoudre ces deux problèmes, nous proposons des algorithmes heuristiques inspiré par ALG-SFSB et ALG-SFMB. La conception de ces algorithmes est basée sur trois étapes. Dans la première étape, la solution d'allocation de fréquences est obtenue par une approche gloutonne. Dans la deuxième étape, l'ordonnancement des utilisateurs pour une seule station de base est dérivé en modifiant ALG-SFSB. Enfin, la troisième étape consiste à trouver l'association des utilisateurs en appliquant de manière itérative les deux premières étapes. Nous considérons d'abord OAFED-MFSB et décrivons un algorithme glouton en modifiant ALG-SFSB. Ensuite, nous considérons OAFED-MFMB en proposant un algorithme itératif.

OAFED-MFSB

Pour résoudre OAFED-MFSB, nous proposons un algorithme glouton, appelé ALG-MFSB, qui alloue les fréquences aux utilisateurs qui nécessitent le plus petit nombre de slots. En d'autres termes, à l'utilisateur u est alloué la fréquence f_u telle que

$$f_u = \arg \min_{f \in \mathcal{F}} v_u^f. \quad (4.19)$$

En cas d'égalité, à l'utilisateur u est alloué la fréquence la moins utilisée (celle qui est allouée aux d'utilisateurs le plus petit nombre de fois). Une fois que l'allocation de fré-

quences est obtenue, ALG-MFSB ordonnance les utilisateurs en appliquant une version modifiée de ALG-SFSB. L'ordonnanceur Σ est maintenant noté par une matrice $[\sigma_{s,f}]$ où $\sigma_{s,f} = u$ si l'utilisateur u est ordonné pendant le *slot* s sur la fréquence f . Par conséquent, ALG-MFSB (y compris la procédure de re-ordonnement) fonctionne comme ALG-SFSB sauf qu'il utilise la matrice $[\sigma_{s,f}]$ et doit vérifier que deux utilisateurs ne peuvent pas être ordonnés pendant le même *slot* en utilisant la même fréquence.

OAFED-MFMB

Afin de résoudre OAFED-MFMB, nous appliquons de manière itérative ALG-MFSB dans chaque station de base. À chaque itération, deux étapes principales sont exécutées : (i) la station de base qui maximise le nombre d'utilisateurs servis est sélectionnée et (ii) les utilisateurs servis sont supprimés de l'ensemble de tous les utilisateurs. L'idée de cet algorithme itératif (appelé ALG-MFMB) est similaire à celle de ALG-SFMB. ALG-MFMB est décrit dans le pseudo-code de l'algorithme 11. Nous utilisons les mêmes notations que précédemment.

Algorithme 11 L'algorithme ALG-MFMB

```

1: fonction MFMB( $\mathcal{U}, \mathcal{B}, \mathcal{F}, \mathbf{v}, \mathbf{A}$ )
2:    $\mathcal{X} \leftarrow \mathcal{U}$ 
3:    $\mathcal{Y} \leftarrow \mathcal{B}$ 
4:    $\mathcal{Z} \leftarrow \emptyset$ 
5:    $\mathcal{A} \leftarrow \emptyset$ 
6:   tant que  $\mathcal{X} \neq \emptyset$  et  $\mathcal{Y} \neq \emptyset$  faire
7:     pour  $b$  dans  $\mathcal{Y}$  faire
8:        $(\Sigma, \mathcal{A}_b) \leftarrow \text{ALG-MFSB}(\mathcal{X}, \mathbf{d}, \mathbf{A}_b, \mathbf{v}_b)$ 
9:     fin pour
10:     $b^* \leftarrow \arg \max \{\mathcal{A}_y\}$ 
11:     $\mathcal{Y} \leftarrow \mathcal{Y} \setminus \{b^*\}$ 
12:     $\mathcal{X} \leftarrow \mathcal{X} \setminus \mathcal{A}_{b^*}$ 
13:     $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{b^*\}$ 
14:  fin tant que
15:  retourner  $\mathcal{A}_{\mathcal{Z}}$ 
16: fin fonction

```

ALG-MFMB appelle la fonction MFSB au niveau de la ligne 8 où cette dernière trouve l'allocation de fréquences selon (4.19). La complexité asymptotique de ALG-MFSB est $\mathcal{O}(\ell + k \log k + km(k + \ell m)) = \mathcal{O}(k^2 m + k \ell m^2)$. Par conséquent, la complexité asymptotique de ALG-MFMB est $\mathcal{O}(nq(k^2 m + k \ell m^2))$ où $q \triangleq \min\{k, n\}$.

Bien sûr, puisque OAFED-MFSB est NP-difficile, donc ALG-MFSB et ALG-MFMB ne garantissent pas de trouver une solution optimale. De plus, ALG-MFMB n'est probablement pas un algorithme d'approximation puisque ALG-MFSB n'est pas optimal pour OAFED-MFSB. Néanmoins, notez que la stratégie gloutonne utilisée pour allouer les fréquences aux utilisateurs peut être considérée comme une des meilleures stratégies, c'est-à-dire, pour l'utilisateur u , il est optimal de choisir la fréquence qui se caractérise par le moins grand nombre de *slots* afin de satisfaire sa demande.

4.6.3 Résultats de simulations

Dans cette section, nous présentons les résultats de simulation pour illustrer les performances des algorithmes proposés. Nous considérons une zone géographique modélisée par un carré de côté égal à 20 mètres dans laquelle les utilisateurs et les stations de base sont distribués de manière aléatoire et uniforme. Les gains des canaux sont basés sur les spécifications 3GPP (3GPP, 2010) et sont modélisés de la même façon que (Zhuang *et al.*, 2016). L'arrivée d'énergie $E_{b,s}$ suit une distribution de Poisson avec le paramètre λ . Sauf indication contraire, les paramètres de simulations sont donnés sur le tableau 4.2. Le problème d'optimisation (4.17) est modélisé en Python avec PuLP (Mitchell *et al.*, 2011) et résolu en utilisant le solveur CPLEX (IBM, 2010). Les simulations sont moyennées sur 1000 réalisations.

La figure 4.9 montre que lorsque λ augmente, le nombre d'utilisateurs servis augmente rapidement pour atteindre 10. En effet, pour un k fixe, l'écart au niveau de l'axe des ordonnées entre $\lambda = 0,5$ et $\lambda = 1$ est plus grand que l'écart entre $\lambda = 1$ et $\lambda = 2$. Ceci

Paramètre	Valeur
Puissance de transmission	30 dBm
Bande passante B	20 MHz
Fréquence porteuse	2 GHz
Puissance du bruit	-174 dBm/Hz
Gain du canal entre u et b	$30,6 + 36,7 \log_{10} d_{b \rightarrow u}$
Taux d'arrivée de l'énergie	$\lambda = 0,5$
Taille des données	$w_u \sim \text{unif}\{1Kb, 1Mb\}$
Nombre de <i>slots</i>	$m = 10$
Dates limites	$d_u \sim \text{unif}\{1, m\}$
Nombre de stations de base	$n = 10$
Nombre de <i>slots</i>	$m = 10$

Tableau 4.2 Paramètres de simulations pour OAFED.

implique que lorsque l'énergie est toujours disponible alors le problème devient équivalent à ordonnancer les utilisateurs par ordre croissant d'énergie requise, ce qui est une tâche simple.

La figure 4.10 montre que lorsque m augmente, plus d'utilisateurs peuvent être servis. En outre, puisque l'énergie est fixe, alors le nombre d'utilisateurs servis ne peut pas augmenter au delà d'une certaine valeur de k même si m augmente davantage, ce qui est montré par la saturation des performances lorsque k augmente.

La figure 4.11 montre que ALG-SFMB sert un nombre d'utilisateurs proche de celui de OPT-SFMB même pour des grandes valeurs de k . En outre, au fur et à mesure que λ augmente, ALG-SFMB se rapproche de OPT-SFMB. La performance d'ALG-SFMB est meilleure que celle donnée par le facteur d'approximation de 2. En effet, le rapport entre o et a est de 1,082 pour $k = 50$ et $\lambda = 0,5$.

La figure 4.12 montre aussi qu'ALG-SFMB et OPT-SFMB sont proches l'un de l'autre. Lorsque m devient plus grand et lorsque k et n sont fixes, l'écart entre les deux algorithmes est réduit. Nous pouvons voir aussi qu'ALG-SFMB a une performance supérieure

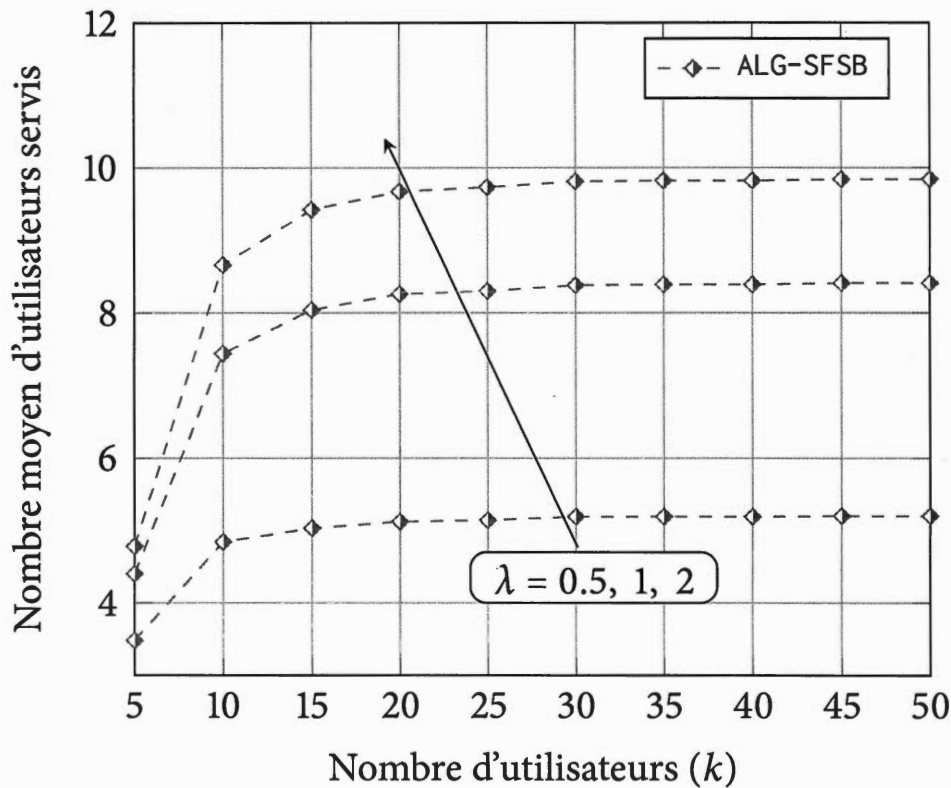


Figure 4.9 Nombre moyen d'utilisateurs servis par ALG-SFSB en fonction de k pour OAFED-SFSB.

à celle de la garantie de performance théorique de 2-approximation.

La figure 4.13 montre que, pour n petit, ALG-SFMB se rapproche de OPT-SFMB lorsque k augmente. Nous pouvons voir que plus les stations de base sont disponibles, alors plus d'utilisateurs sont servis par les deux algorithmes et, heureusement, ALG-SFMB se rapproche de OPT-SFMB. L'écart entre les deux algorithmes reste faible, même pour des grandes valeurs de k et n , ce qui illustre l'efficacité d'ALG-SFMB.

La figure 4.14 montre qu'ALG-MFMB est proche de l'optimal. Lorsque le nombre de stations de base augmente, plus d'utilisateurs sont servis même lorsque l'énergie est constante. Nous pouvons voir aussi que le rapport (OPT-MFMB-sur-ALG-MFMB) des uti-

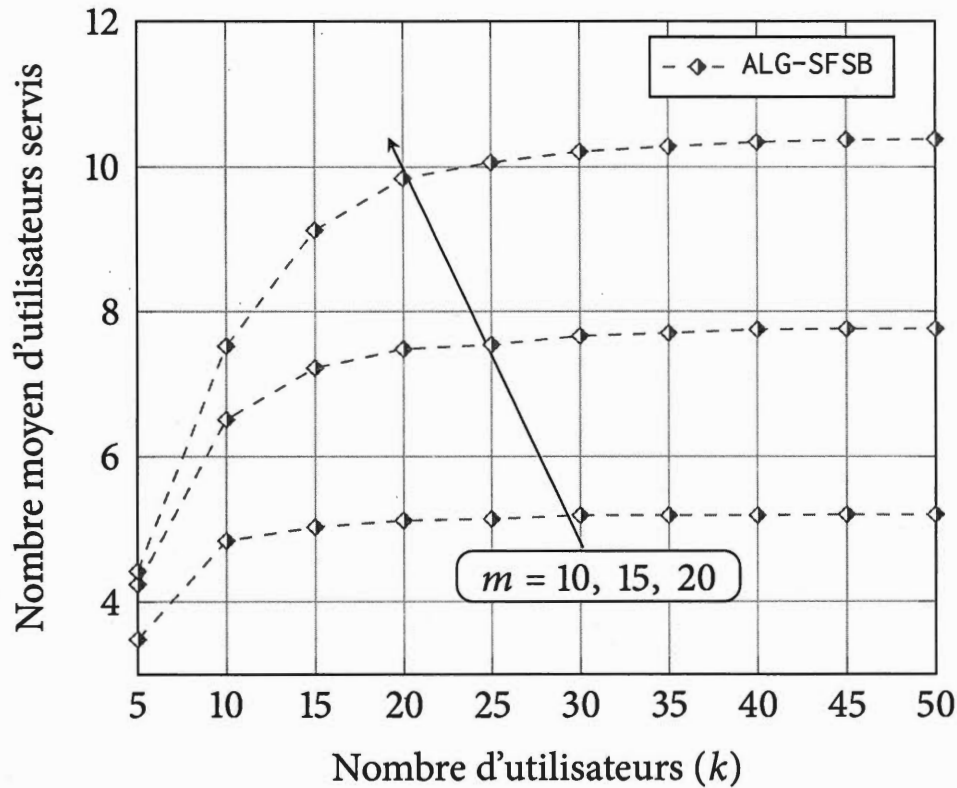


Figure 4.10 Nombre moyen d'utilisateurs servis par ALG-SFSB en fonction de k pour OAFED-SFSB.

lisseurs servis est de 1,136 pour $k = 20$ et $n = 1$. Ce rapport devient égal à 1,075 pour $k = 20$ et $n = 4$. Par conséquent, nous concluons que le rapport de performance diminue lorsque le nombre de stations de base augmente, ce qui illustre l'efficacité d'ALG-MFMB dans les réseaux DCNs.

La figure 4.15 montre qu'ALG-MFMB sert plus d'utilisateurs lorsque λ , n ou ℓ augmente. Il est important de noter que le nombre d'utilisateurs servis par ALG-MFMB augmente plus rapidement avec n qu'avec ℓ . Par exemple, d'une part, pour $\lambda = 1$ et lorsque n varie de 6 à 10, ALG-MFMB sert 1,45 fois plus d'utilisateurs pour $\ell = 2$ et sert 1,42 fois plus d'utilisateurs pour $\ell = 10$. Par contre, pour $\lambda = 1$ et lorsque ℓ varie de 2 à 10, alors ALG-MFMB sert 1,035 fois plus d'utilisateurs pour $n = 6$ et sert 1,011 fois plus d'utilisateurs pour $n =$

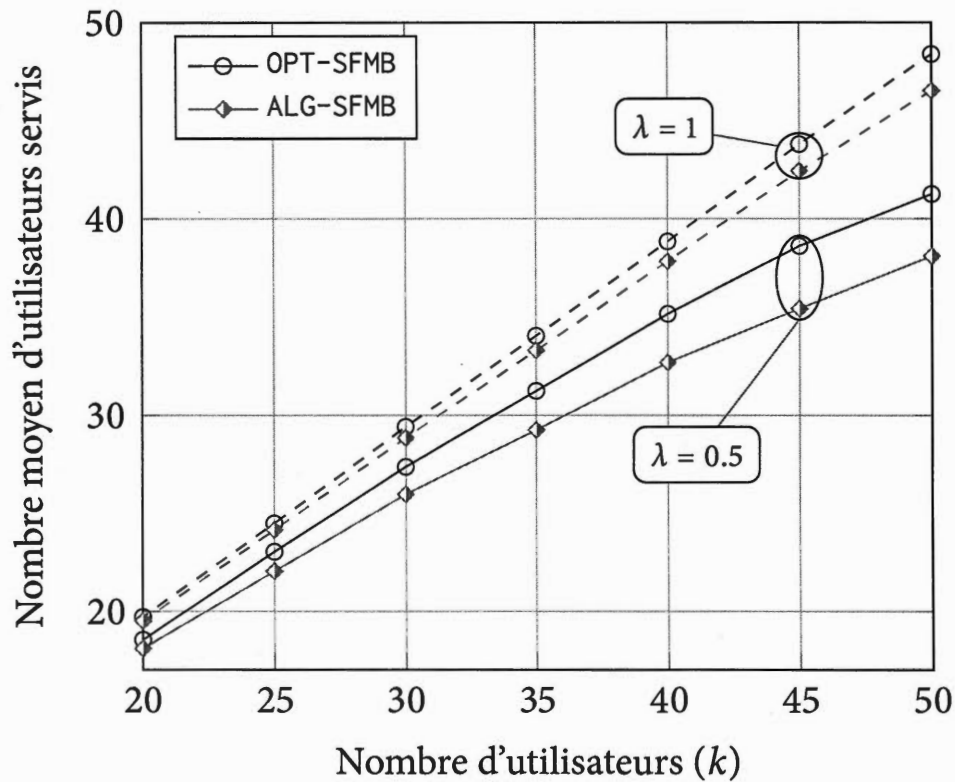


Figure 4.11 Nombre moyen d'utilisateurs servis par ALG-SFMB et OPT-SFMB en fonction de k pour OAFED-SFMB pour différents λ .

10. Nous pouvons conclure que (i) avoir plus de canaux est avantageux (en termes de maximisation du nombre d'utilisateurs) lorsque λ est grand et n est fixe, et (ii) avoir plus de stations de base est avantageux lorsque λ est petit et ℓ est fixe.

4.7 Le problème OAED-ON

Dans cette partie, nous étendons le modèle de système de PAR dans le scénario *online*. Dans ce modèle, il existe T trames (au lieu d'une seule) de m slots chacune. Au début de chaque trame, t , un ensemble $\mathcal{U}_t$ de k_t utilisateurs arrivent. Les utilisateurs arrivés au cours de la trame t sont pris en compte au début de la trame suivante $t + 1$. La décision

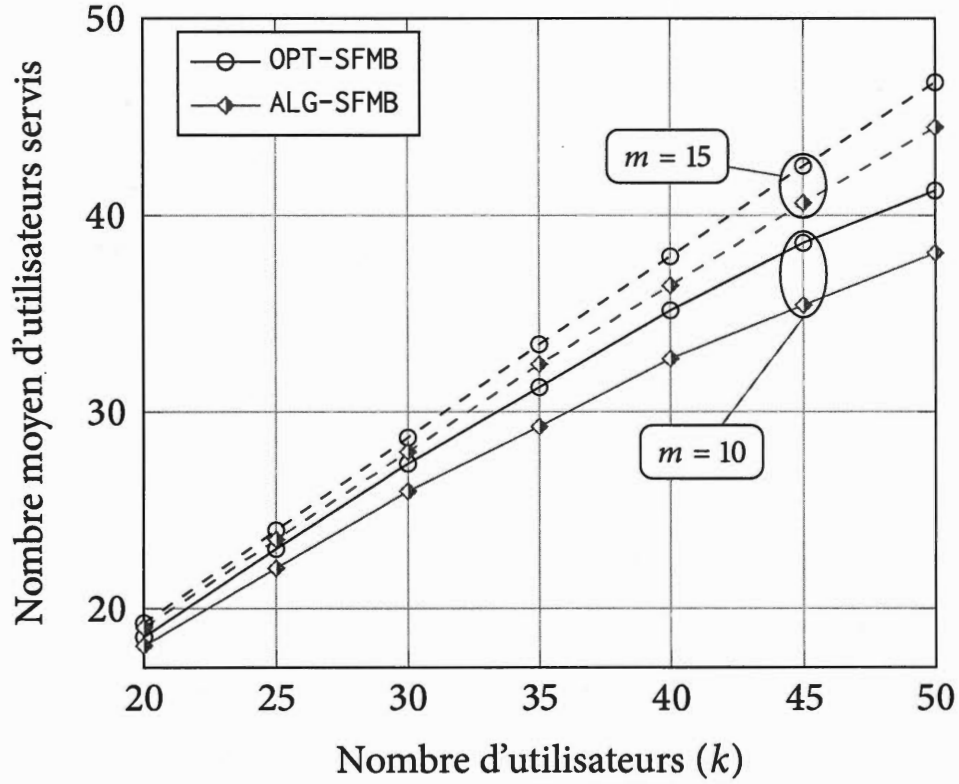


Figure 4.12 Nombre moyen d'utilisateurs servis par ALG-SFMB et OPT-SFMB en fonction de k pour OAFED-SFMB pour différents m .

est prise au début de chaque trame d'une façon irrévocable. La date limite de l'utilisateur $u \in \mathcal{U}_t$ est donnée par $d_u = tT$, c.-à-d. u doit télécharger ses données avant la fin de sa trame. La quantité d'énergie $E_{b,t}$ récoltée par la station de base b pendant la trame t est stockée dans la batterie de b de capacité E_{max} . Comme précédemment, nous normalisons $E_{b,t}$ par p_b pour obtenir $A_{b,t} \triangleq E_{b,t}/p_b$ qui représente le nombre minimum de *slots* pouvant être utilisés par b pour transmettre avec la puissance p_b pendant la trame t . L'arrivée des utilisateurs et d'énergie est arbitraire et ne suit aucune distribution particulière. En d'autres termes, l'entrée du problème est supposée être non stochastiques comme dans (Shen *et al.*, 2016). Dans cette partie, l'allocation de fréquence universelle est supposée, c.-à-d., $\mathcal{F} = \{1\}$. Les gains des canaux sans fil, les rapports SINR et les débits

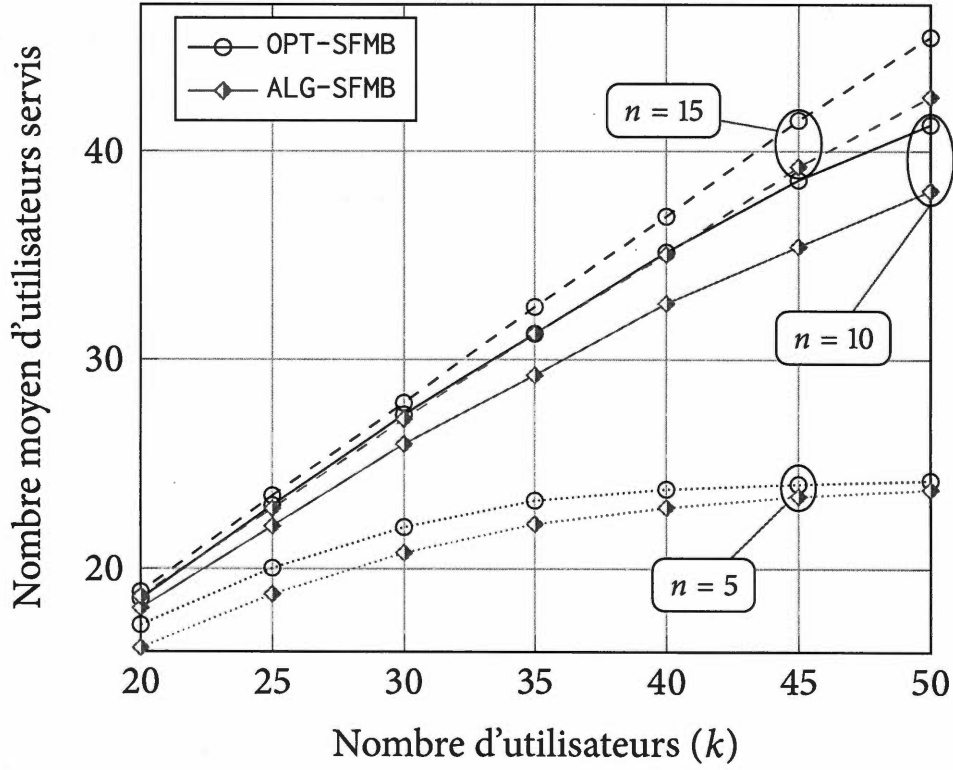


Figure 4.13 Nombre moyen d'utilisateurs servis par ALG-SFMB et OPT-SFMB en fonction de k pour OAFED-SFMB pour différents n .

de transmission changent d'une trame à une autre mais sont fixes pour la durée d'une trame.

En remplaçant l'indice f dans (4.1) par l'indice de la trame t , nous pouvons calculer les rapports SINR, les débits de transmission, ainsi que le nombre de *slots* requis par l'utilisateur $u \in \mathcal{U}_t$ quand il est associé à la station de base b , respectivement, comme suit :

$$\text{SINR}_{b \rightarrow u}^t = \frac{p_b g_{b \rightarrow u}^t}{\sigma^2 + \sum_{b' \in \mathcal{B} \setminus \{b\}} p_{b'} g_{b' \rightarrow u}^t}, \quad (4.20)$$

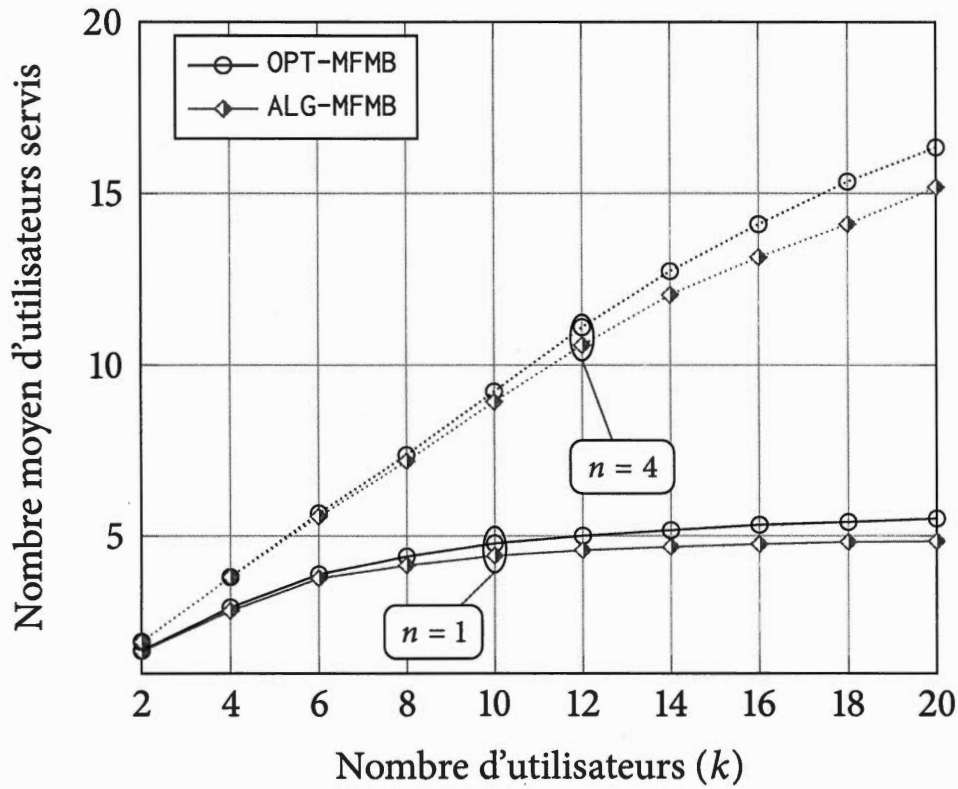


Figure 4.14 Nombre moyen d'utilisateurs servis par ALG-MFMB et OPT-MFMB en fonction de k pour OAFED-MFMB pour différents n ($\ell = 2$).

$$R_{b \rightarrow u}^t = \log_2 (1 + \text{SINR}_{b \rightarrow u}^t), \quad (4.21)$$

$$v_{b \rightarrow u}^t \triangleq \left\lceil \frac{w_u}{\tau B R_{b \rightarrow u}^t} \right\rceil. \quad (4.22)$$

4.7.1 Le cas d'une seule station de base

Dans cette partie, nous étudions OAED-ON dans le cas où $n = 1$. Ce problème est appelé ordonnancement avec énergie et dates limites *online* (OED-ON). Tous les indices faisant référence à b sont supprimés. OED-ON peut être formulé comme le programme

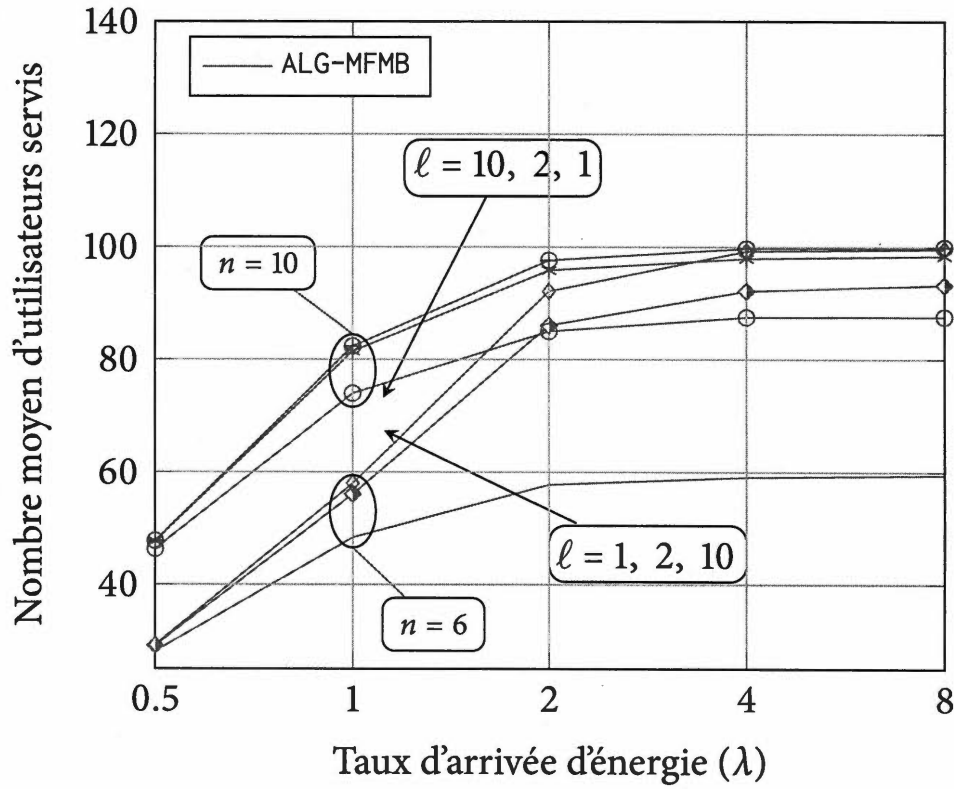


Figure 4.15 Nombre moyen d'utilisateurs servis par ALG-MFMB pour OAFED-MFMB en fonction de λ pour différents n et ℓ ($k = 100$).

linéaire en nombres entiers suivant :

$$\text{maximiser} \quad \sum_{t=1}^T \sum_{u \in \mathcal{U}_t} x_{u,t} \quad (4.23a)$$

$$\text{sous les contraintes} \quad \sum_{u \in \mathcal{U}_t} v_u^t x_{u,t} \leq z_t, \forall t, \quad (4.23b)$$

$$z_t \leq m, \forall t, \quad (4.23c)$$

$$\sum_{s=1}^t z_s \leq \min\left\{\sum_{s=1}^t A_s, A_{\max}\right\}, \forall t, \quad (4.23d)$$

où, $x_{u,t} = 1$ si et seulement si u est ordonnancé pendant la trame t et $z_t \geq 0$ représente la quantité d'énergie utilisée pendant la trame t et $A_{\max}$ est la capacité en slots qui corres-

pond à E_{max} .

Afin de résoudre OED-ON dans le scénario *online*, d'abord, nous proposons l'algorithme *offline* suivant. Cet algorithme est appelé OFF-OED et il est défini lorsque l'ensemble des utilisateurs ainsi que l'énergie sont connus a priori. Le pseudo-code de OFF-OED est donné dans l'algorithme 12. Soit $\mathbf{A} = [A_1, \dots, A_T]$ le vecteur d'arrivée d'éner-

Algorithme 12 L'algorithme OFF-OED

```

1: fonction OFF( $\mathcal{U}, \mathbf{A}, m, T, \mathbf{v}$ )
2:    $\mathcal{A}_t \leftarrow \emptyset$ 
3:    $m_t \leftarrow m$  pour tout  $t$ 
4:   Trier les utilisateurs dans  $\mathcal{U}$  selon SMR
5:   pour  $i$  dans  $\mathcal{U}$  faire
6:     Soit  $(u, t)$  la paire qui correspond à  $i$ 
7:      $C_t \leftarrow \sum_{s=1}^t A_s$ 
8:     si  $v_u^t \leq \min\{C_t, m_t\}$  alors
9:        $\mathcal{A}_t \leftarrow \mathcal{A}_t \cup \{u\}$ 
10:       $m_t \leftarrow m_t - v_u^t$ 
11:      Mettre à jour  $\mathbf{A}$  // La mise à jour de  $\mathbf{A}$  : (i) diminue l'énergie né-
        cessaire pour ordonnancer un utilisateur, ou (ii) rajoute l'énergie utilisée pour or-
        donnancer un ancien utilisateur qui a été supprimé.
12:    fin si
13:  fin pour
14:  retourner  $\{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_T\}$ 
15: fin fonction

```

gie (*slots*) et $\mathbf{v} = [v_u^t]$ la matrice de *slots* requis. OFF-OED itère l'ensemble des utilisateurs $\mathcal{U} \triangleq \bigcup_{t \in \{1, \dots, T\}} \mathcal{U}_t$ triés par ordre croissant de leur nombre requis de *slots*. Nous appelons cette règle de tri comme *slots moins requis* (SMR). Pour chaque utilisateur $i \in \mathcal{U}$, OFF-OED l'ordonnance à t si (i) il y a assez d'énergie, et (ii) sa date limite n'est pas encore atteinte. Après, OFF-OED met à jour l'énergie en itérant les trames à partir de t dans le sens inverse et décrémente v_u^t unités d'énergie de $\mathbf{A}$.

Théorème 4.7.1 OFF-OED est un algorithme *offline* polynomial optimal pour OED-ON.

Démonstration : la preuve est de montrer que n'importe quel algorithme *offline* op-

timal peut être modifié pour se comporter comme OFF-OED sans diminuer le nombre d'utilisateurs ordonnancés. Soit OPT-OED un algorithme *offline* optimal pour OED-ON et soit $\mathcal{U} = \{1, 2, \dots, k\}$ où $k \triangleq \sum_{t=1}^T k_t$. Nous montrons que pour chaque utilisateur $u \in \mathcal{U}$, il est possible de construire un algorithme *offline* OPT-OED_{*u*} qui vérifie les trois propriétés suivantes : (i) OPT-OED_{*u*} ordonnance les premiers $u - 1$ utilisateurs exactement comme le fait OPT-OED; (ii) si l'ordonnancement de l'utilisateur u ne respecte pas les contraintes des dates limites ou d'énergie, alors OPT-OED_{*u*} supprime l'utilisateur ordonnancé nécessitant le plus grand nombre de *slots*; et (iii) $o_u \geq o$, où o_u et o représentent le nombre d'utilisateurs ordonnancés par OPT-OED et OPT-OED_{*u*}, respectivement.

Supposons qu'après l'ordonnancement de u , OPT-OED et OPT-OED_{*u*} ordonnancent $\mathcal{A} \cup \{i\}$ et $\mathcal{A} \cup \{j\}$, respectivement, où $\mathcal{A}$ est un ensemble commun d'utilisateurs ordonnancés et $i \neq j$. Supposons que i (resp. j) est ordonnancé à la trame t (resp. t'). Si $v_i^t \geq v_j^{t'}$, alors les trois propriétés (i)-(iii) sont trivialement vraies. Si $v_i^t \leq v_j^{t'}$, alors OPT-OED_{*u*} peut supprimer j (ou tout autre utilisateur de $\mathcal{A}$ nécessitant au moins $v_j^{t'}$) et ordonnance i à la place. De cette manière, les propriétés sont encore vraies.

Nous pouvons appliquer la procédure précédente k fois comme suit. Étant donné OPT-OED, nous l'appliquons pour le premier utilisateur de $\mathcal{U}$ afin d'obtenir OPT-OED₁. Étant donné OPT-OED₁, nous l'appliquons pour le second utilisateur afin d'obtenir OPT-OED₂, et ainsi de suite. Évidemment, OPT-OED_{*k*} fonctionne de la même manière que OFF-OED. Puisque OFF-OED est un algorithme polynomial, le théorème est prouvé. ■

Pour résoudre OED-ON, nous proposons un algorithme *online* simple appelé ON-OED. Cet algorithme est basé sur une approche gloutonne déterministe. Il fonctionne principalement comme suit. Chaque fois qu'un ensemble d'utilisateurs $\mathcal{U}_t$ arrive, il trie les utilisateurs déjà arrivés selon SMR et ordonnance le nombre maximal d'utilisateurs en utilisant toute (ou le maximum de) l'énergie disponible. ON-OED est illustré dans le pseudo-code de l'algorithme 13.

Algorithme 13 L'algorithme ON-OED appliqué pendant trame t

```

1:  $\mathcal{A}_t \leftarrow \emptyset$ 
2:  $m_t \leftarrow m$ 
3:  $C_t \leftarrow \sum_{s=1}^t A_s$ 
4: Trier les utilisateurs de  $\mathcal{U}_t$  selon SMR
5: pour  $u$  dans  $\mathcal{U}_t$  faire
6:   si  $v_u^t \leq \min\{C_t, m_t\}$  alors
7:      $\mathcal{A}_t \leftarrow \mathcal{A}_t \cup \{u\}$ 
8:      $m_t \leftarrow m_t - v_u^t$ 
9:     Mettre à jour  $A$ 
10:     $C_t \leftarrow \sum_{s=1}^t A_s$ 
11:   fin si
12: fin pour

```

Dans ce qui suit, nous prouvons que ON-OED est m -compétitif.

Théorème 4.7.2 ON-OED est un algorithme polynomial m -compétitif.

Démonstration : considérons une instance arbitraire de OED-ON. Soit A_t (resp. o_t) le nombre d'utilisateurs ordonnancés pendant la trame t par ON-OED (resp. par OFF-OED). Nous notons par $v(A_t)$ (resp. $v(o_t)$) le nombre de *slots* utilisés pendant la trame t par ON-OED (resp. par OFF-OED). L'énergie disponible restante L_t pendant la trame $t \geq 2$ pour ON-OED est donnée par :

$$L_t = L_{t-1} + A_t - v(A_{t-1}), \quad (4.24)$$

où $L_1 = A_1$. Après avoir résolu la récurrence (4.24), nous obtenons :

$$L_{T+1} = \sum_{t=1}^{T+1} A_t - \sum_{t=2}^{T+1} v(A_{t-1}) = \sum_{t=1}^{T+1} A_t - v(\text{ON}), \quad (4.25)$$

où $v(\text{ON})$ et $v(\text{OFF})$ représentent le nombre de *slots* utilisés par ON-OED et par OFF-OED, respectivement, pendant les T trames. Puisque $v(\text{OFF}) \leq \sum_{t=1}^T A_t$, alors, en utili-

sant (4.25), nous obtenons :

$$v(\text{OFF}) \leq L_{T+1} - A_{T+1} + v(\text{ON}) \leq L_{T+1} + v(\text{ON}). \quad (4.26)$$

Maintenant, nous montrons que (i) $v(\text{OFF}) \geq 0$ et (ii) $v(\text{ON}) \leq mA$, où $0 = \sum_{t=1}^T 0_t$ et $A = \sum_{t=1}^T A_t$. En effet, en observant que chaque utilisateur ordonnancé nécessite au moins un *slot* et au plus m *slots*, alors (i) et (ii) sont vraies. Nous concluons que $0 \leq mA + L_{T+1}$. Étant donné qu'ON-OED utilise d'une manière gourmande l'énergie disponible, alors l'énergie restante L_{T+1} sera toujours bornée. En effet, la capacité de la batterie de la station de base est limitée, donc nous pouvons affirmer que $L_{T+1} \leq A_{\max}$. Nous concluons qu'il existe une constante $\alpha = A_{\max}$ telle que pour toutes les instances de OED-ON, nous avons $0 \leq cA + \alpha$, où $c = m$. Par conséquent, ON-OED est m -compétitif. Il est clair qu'ON-OED est un algorithme polynomial. Ceci termine la démonstration du théorème. ■

Remarque 4.7.3 (ON-OED est strictement m -compétitif) Nous pouvons montrer que ON-OED est en fait strictement m -compétitif. Les utilisateurs ordonnancés $\mathcal{A}_{\text{ON}}$ retournés par ON-OED peuvent être divisés en deux ensembles disjoints $\mathcal{A}_{\text{ON}} = \mathcal{C} \cup \mathcal{D}_{\text{ON}}$, où $\mathcal{C}$ contient les utilisateurs ordonnancés à la fois par OFF-OED et ON-OED et $\mathcal{D}_{\text{ON}}$ contient les utilisateurs ordonnancés par ON-OED uniquement. De même, $\mathcal{A}_{\text{OFF}} = \mathcal{C} \cup \mathcal{D}_{\text{OFF}}$. Soient $A = |\mathcal{A}_{\text{ON}}|$, $0 = |\mathcal{A}_{\text{OFF}}|$, $C = |\mathcal{C}|$, $D_{\text{ON}} = |\mathcal{D}_{\text{ON}}|$, et $D_{\text{OFF}} = |\mathcal{D}_{\text{OFF}}|$. Chaque fois qu'ON-OED ordonnance un utilisateur u dans $\mathcal{D}_{\text{ON}}$, il perd au plus $v_u - 1$ utilisateurs. En d'autres termes, si ON-OED avait ordonné un utilisateur de $\mathcal{D}_{\text{OFF}}$ au lieu de $u \in \mathcal{D}_{\text{ON}}$, il aurait ordonné au maximum $v_u - 1$ plus d'utilisateurs. Par conséquent, $D_{\text{OFF}} \leq mD_{\text{ON}}$ et donc $0 \leq mD_{\text{ON}} + C$. Nous savons que $D_{\text{ON}} = A - C$. Ainsi, $0 \leq m(A - C) + C \leq mA$. Donc, ON-OED est strictement m -compétitif.

Remarque 4.7.4 (Une limite supérieure étroite) Nous donnons une instance de OED-ON pour laquelle ON-OED atteint un ratio compétitif de m . Soit $T = 2q$ pour un entier positif q . L'arrivée d'énergie est $A_t = m$ si t est impair et 0 sinon. Pour chaque trame t , il existe un seul utilisateur ou m utilisateurs, c'est-à-dire que $\mathcal{U}_t = \{1\}$ si t est impair et

$\{1, 2, \dots, m\}$ sinon, où $v_1^t = m$ si t est impair et $v_u^t = 1$, pour $u \in \{1, 2, \dots, m\}$, sinon.

ON-OED ordonnance un utilisateur dans chaque trame impaire et aucun utilisateur dans les trames paires. Par conséquent, au total, ON-OED ordonnance q utilisateurs. Cependant, OFF-OED conserve l'énergie dans les trames impaires et l'utilise dans les trames paires. Autrement dit, OFF-OED n'ordonnance aucun utilisateur dans les trames impaires et m utilisateurs dans chaque trame paire. Au total, OFF-OED ordonnance qm utilisateurs. Le ratio, offline-sur-online, est égal à m . Cela montre qu'ON-OED atteint le meilleur ratio compétitif possible.

Puisque l'entrée d'OED-ON est donnée par un adversaire qui tente de pénaliser autant que possible l'algorithme *online* ON-OED, ce dernier est confronté au compromis d'exploration (économiser l'énergie) et d'exploitation (utilisation d'énergie). Afin de résoudre ce compromis, nous abordons OED-ON dans le cadre d'apprentissage *online* (*online learning* en anglais).

L'apprentissage *online* nous a permis de développer un algorithme d'apprentissage incrémental efficace. Cet algorithme est noté par ONL-OED. Il permet de minimiser le *regret*, qui est défini, de manière informelle, comme la différence entre les récompenses cumulées et la récompense du meilleur algorithme. Cette approche compare les performances des algorithmes en utilisant une méthode additive plutôt qu'une méthode multiplicative utilisée dans l'analyse compétitive précédent. ONL-OED est basé sur l'approche du bandit manchot (*Multi-Armed Bandit (MAB)* en anglais) (Auer *et al.*, 2002). Dans MAB, un joueur doit choisir un bras parmi un ensemble de machines non identiques dans une séquence de tours afin de maximiser sa récompense (Auer *et al.*, 2002). MAB a fait l'objet d'une attention considérable parce qu'il modélise de manière simple le compromis entre l'exploration et l'exploitation. Dans OED-ON, la station de base est confrontée au dilemme d'explorer plus d'utilisateurs et d'énergie ou de les exploiter pour ordonnancer beaucoup d'utilisateurs à chaque trame.

Nous modélisons OED-ON en utilisant MAB comme suit. La station de base représente le bandit, qui implémente ONL-OED. Pour spécifier les bras, une approche simple consisterait à définir les unités d'énergie disponibles dans chaque trame comme un bras, c'est-à-dire que, si L_t est l'énergie disponible dans la trame t , les bras sont $\{0, 1, \dots, L_t\}$. De cette manière, cette approche nécessite que le nombre de bras soit variable. MAB avec un nombre variable de bras est connu dans la littérature sous le nom de MAB mortel stochastique (*Mortal Multi-Armed Bandit (MMAB)* en anglais) (Chakrabarti *et al.*, 2008). MMAB est principalement étudié dans le cas stochastique (Shen *et al.*, 2016), ce qui le rend différent de notre problème OED-ON, qui est non-stochastique. Par conséquent, afin de pouvoir utiliser MAB, nous adoptons une approche différente en divisant l'énergie disponible en plusieurs niveaux. Pour ce faire, soit δ un nombre positif appartenant à l'intervalle $(0, 100]$. Nous définissons le bras i comme celui qui correspond au pourcentage $(i - 1)\delta$ de l'énergie disponible, où $i \in \{1, 2, \dots, 1 + \lfloor 100/\delta \rfloor\}$. Soit $\Delta \triangleq 1 + \lfloor 100/\delta \rfloor$ le nombre de bras disponibles. Enfin, après avoir choisi le bras $\kappa_t \in \{1, 2, \dots, \Delta\}$, ONL-OED reçoit une récompense $r_{\kappa_t}(t)$ représentant le nombre d'utilisateurs ordonnancés. La récompense moyenne de ONL-OED après T trames est définie comme suit :

$$\mathbb{E}[r_{\text{ONL}}(T)] \triangleq \sum_{t=1}^T \mathbb{E}[r_{\kappa_t}(t)], \quad (4.27)$$

où $\mathbb{E}[\cdot]$ représente l'espérance mathématique qui calcule la moyenne sur les choix aléatoires d'ONL-OED. De même, nous définissons la récompense globale d'un adversaire ADV-OED après T trames comme :

$$r_{\text{ADV}}(T) \triangleq \sum_{t=1}^T r_{\kappa_t^*}(t), \quad (4.28)$$

où $r_{\kappa_t^*}(t)$ est la récompense obtenue par ADV-OED en choisissant le bras κ_t^* dans la trame

t . Enfin, le *regret* $\mathfrak{R}_{\text{ONL}}(T)$ de ONL-OED après T trames est défini comme suit :

$$\mathfrak{R}_{\text{ONL}}(T) \triangleq r_{\text{ADV}}(T) - \mathbb{E}[r_{\text{ONL}}(T)]. \quad (4.29)$$

Ainsi, le regret mesure combien ONL-OED gagne ou perd en choisissant $\kappa_1, \kappa_2, \dots, \kappa_T$ au lieu de $\kappa_1^*, \kappa_2^*, \dots, \kappa_T^*$.

Nous nous concentrons sur deux types d'adversaires déterministes. Le premier, appelé Meilleur Bras Individuel (MBI-OED), choisit le meilleur pourcentage d'énergie pendant l'horizon temporel T . Le second, appelé Meilleur Bras Maximum (MBM-OED) choisit toujours 100% d'énergie. Nous notons que MBM-OED est identique à ON-OED. Si ADV-OED = MBI-OED, alors (4.28) devient :

$$r_{\text{MBI}}(T) \triangleq \max_{i \in \{1, \dots, \Delta\}} \sum_{t=1}^T r_i(t). \quad (4.30)$$

Dans ce cas, le regret définit dans (4.29) est appelé *regret faible* et est donné par :

$$\mathfrak{R}_{\text{ONL}}^w(T) \triangleq r_{\text{MBI}}(T) - \mathbb{E}[r_{\text{ONL}}(T)]. \quad (4.31)$$

D'autre part, si ADV-OED = MBM-OED, alors (4.28) devient :

$$r_{\text{MBM}}(T) \triangleq \sum_{t=1}^T r_{\Delta}(t), \quad (4.32)$$

et dans ce cas, le regret définit dans (4.29) est appelé *regret souple* et est donné par :

$$\mathfrak{R}_{\text{ONL}}^s(T) \triangleq r_{\text{MBM}}(T) - \mathbb{E}[r_{\text{ONL}}(T)]. \quad (4.33)$$

L'algorithme d'apprentissage ONL-OED est basé sur la méthode de pondération exponentielle pour l'exploration et l'exploitation (EXP3). ONL-OED calcule, pour chaque bras

i , un poids $w_i(t)$ dans la trame t . Ces poids sont utilisés pour construire une distribution de probabilité $\pi(t) = [\pi_1(t), \pi_2(t), \dots, \pi_\Delta(t)]$ sur les bras dans chaque trame t . Cette distribution est construite en utilisant un paramètre de réglage θ afin d'obtenir un mélange de deux distributions—une distribution uniforme et une distribution qui attribue à chaque bras une fonction de masse qui est exponentielle dans la récompense estimée (Auer *et al.*, 2002). En se basant sur $\pi(t)$, ONL-OED sélectionne un bras κ_t dans la trame t et reçoit donc une récompense $r_{\kappa_t}(t)$. Le choix aléatoire dans la sélection des bras permet à ONL-OED de tromper l'adversaire.

Plus précisément, ONL-OED choisit le bras i avec la probabilité $\pi_i(t)$ donnée par :

$$\pi_i(t) = (1 - \theta) \frac{w_i(t)}{\sum_{j=1}^{\Delta} w_j(t)} + \frac{\theta}{\Delta}. \quad (4.34)$$

Après avoir sélectionné le bras κ_t en fonction de $\pi(t)$, ONL-OED reçoit la récompense $r_{\kappa_t}(t) \in [0, m]$. Nous normalisons $r_{\kappa_t}(t)$ par m pour qu'il soit dans l'intervalle $[0, 1]$. La récompense normalisée est maintenant égale à $\bar{r}_{\kappa_t}(t) = r_{\kappa_t}(t)/m$. Après cette normalisation, ONL-OED calcule une récompense estimée $\hat{r}_i(t) = [\bar{r}_i(t)/\pi_i(t)] \mathbb{1}_{\{\kappa_t\}}(i)$ où $\mathbb{1}_{\mathcal{A}}(i) = 1$ si et seulement si $i \in \mathcal{A}$. De cette façon, ONL-OED peut compenser les récompenses des autres bras qui sont susceptibles d'être choisis (Auer *et al.*, 2002). Enfin, les poids sont mis à jour comme suit :

$$w_i(t+1) = w_i(t) \exp(\theta \hat{r}_i(t)/\Delta). \quad (4.35)$$

Dans ce qui suit, nous majorons les regrets faible et souple définis dans (4.31) et (4.33), respectivement.

Proposition 4.7.5 *Pour un T donné et un $\theta \in (0, 1]$ donné, le regret faible et le regret*

Algorithme 14 L'algorithme ONL-OED

```

1: Soit  $w_i(1) = 1$  pour tout  $i \in \{1, 2, \dots, \Delta\}$ 
2: Choisir un bras  $\kappa_t$  aléatoirement selon  $\pi(t)$  comme dans (4.34)
3: Recevoir  $r_{\kappa_t}(t)$  et normaliser-le comme  $\bar{r}_{\kappa_t}(t) \leftarrow r_{\kappa_t}(t)/m$ 
4: pour  $i \leftarrow 1$  à  $\Delta$  faire
5:   si  $i = \kappa_t$  alors
6:      $\hat{r}_i(t) \leftarrow \bar{r}_i(t)/\pi_i(t)$ 
7:   sinon
8:      $\hat{r}_i(t) \leftarrow 0$ 
9:   fin si
10:   $w_i(t+1) \leftarrow w_i(t) \exp(\theta \hat{r}_i(t)/\Delta)$ 
11: fin pour
  
```

souple sont respectivement bornés comme suit :

$$\mathfrak{R}_{\text{ONL}}^w(T) \leq \theta(e-1)r_{\text{MBI}}(T) + m \frac{\Delta \ln \Delta}{\theta}, \quad (4.36)$$

$$\mathfrak{R}_{\text{ONL}}^s(T) \leq \theta(1+m(e-2))r_{\text{MBM}}(T) + m \frac{\Delta \ln \Delta}{\theta}. \quad (4.37)$$

Démonstration : nous allons majorer $\mathbb{E}[r_{\text{ONL}}(T)]$. L'idée principale de la preuve vient de (Auer *et al.*, 2002) dans lequel les auteurs majorent le regret faible seulement. Soit $W(t) = \sum_{i=1}^{\Delta} w_i(t)$ et $\zeta \triangleq \theta/\Delta$. Supposons que ONL-OED choisit les bras $\kappa_1, \kappa_2, \dots, \kappa_T$. Nous avons :

$$\begin{aligned}
 \frac{W(t+1)}{W(t)} &\stackrel{(a)}{=} \frac{\sum_{i=1}^{\Delta} w_i(t) \exp(\zeta \hat{r}_i(t))}{W(t)} \\
 &\stackrel{(b)}{=} \sum_{i=1}^{\Delta} \frac{\pi_i(t) - \zeta}{1 - \theta} \exp(\zeta \hat{r}_i(t)) \\
 &\stackrel{(c)}{\leq} \sum_{i=1}^{\Delta} \frac{\pi_i(t) - \zeta}{1 - \theta} \left[1 + \zeta \hat{r}_i(t) + (e-2)(\zeta \hat{r}_i(t))^2 \right] \\
 &\stackrel{(d)}{\leq} 1 + \frac{\zeta}{1 - \theta} r_{\kappa_t}(t)/m + \frac{(e-2)\zeta^2}{1 - \theta} \sum_{i=1}^{\Delta} \hat{r}_i(t)
 \end{aligned}$$

$$\stackrel{(e)}{\leq} \exp \left(\frac{\zeta}{(1-\theta)m} r_{\kappa_t}(t) + \frac{(e-2)\zeta^2}{1-\theta} \sum_{i=1}^{\Delta} \hat{r}_i(t) \right), \quad (4.38)$$

où (a) est due à (4.35) et (b) est due à (4.34). L'inégalité (c) est due à $\zeta \hat{r}_i(t) = \zeta [\bar{r}_i(t)/\pi_i(t)] \mathbb{1}_{\{\kappa_t\}}(i) \leq \zeta/\pi_i(t) \leq 1$ et à $\exp(x) \leq 1 + x + (e-2)x^2$ pour $x \leq 1$. L'inégalité (d) utilise le fait que $\sum_{i=1}^{\Delta} \pi_i(t) \hat{r}_i(t) = r_{\kappa_t}(t)/m$ et $\sum_{i=1}^{\Delta} \pi_i(t) \hat{r}_i(t)^2 = \pi_{\kappa_t}(t) (\bar{r}_{\kappa_t}(t)/\pi_{\kappa_t}(t))^2 \leq \hat{r}_{\kappa_t}(t) = \sum_{i=1}^{\Delta} \hat{r}_i(t)$. Finalement, (e) est due à $1 + x \leq \exp(x)$ pour tout x .

En appliquant le logarithme de deux côtés de (4.38) et en faisant la somme sur t , nous obtenons :

$$\ln \frac{W(T+1)}{W(1)} \leq \frac{\zeta r_{\text{ONL}}(T)}{(1-\theta)m} + \frac{(e-2)\zeta^2}{1-\theta} \sum_{t=1}^T \sum_{i=1}^{\Delta} \hat{r}_i(t). \quad (4.39)$$

Nous savons que $W(T+1) = \sum_{i=1}^{\Delta} w_i(T+1)$ et donc $W(T+1) \geq w_j(T+1)$ pour n'importe quel bras j . En utilisant (4.35), nous pouvons montrer que $w_j(T+1) = \exp(\zeta \sum_{t=1}^T \hat{r}_j(t))$ car $w_j(1) = 1$. Ceci implique que, pour tout bras j

$$\ln \frac{W(T+1)}{W(1)} \geq \ln \frac{\exp(\zeta \sum_{t=1}^T \hat{r}_j(t))}{W(1)} = \zeta \sum_{t=1}^T \hat{r}_j(t) - \ln \Delta. \quad (4.40)$$

En utilisant (4.39) et (4.40), nous obtenons :

$$\zeta \sum_{t=1}^T \hat{r}_j(t) - \ln \Delta \leq \frac{\zeta r_{\text{ONL}}(T)}{(1-\theta)m} + \frac{(e-2)\zeta^2}{1-\theta} \sum_{t=1}^T \sum_{i=1}^{\Delta} \hat{r}_i(t). \quad (4.41)$$

En réarrangeant les termes, nous obtenons :

$$U_{\text{ONL}}(T) \geq m(1-\theta) \left(\sum_{t=1}^T \hat{r}_j(t) - \frac{\ln \Delta}{\zeta} \right) - m(e-2)\zeta \sum_{t=1}^T \sum_{i=1}^{\Delta} \hat{r}_i(t). \quad (4.42)$$

Puisque $\mathbb{E}[\hat{r}_j(t) | \kappa_1, \kappa_2, \dots, \kappa_{t-1}] = \mathbb{E}[\pi_i(t) \bar{r}_j(t)/\pi_i(t) + (1 - \pi_i(t))0] = r_j(t)/m$, nous

obtenons pour tout j :

$$\mathbb{E}[r_{\text{ONL}}(T)] \geq (1 - \theta) \left(\sum_{t=1}^T r_j(t) - m \frac{\ln \Delta}{\zeta} \right) - (e - 2)\zeta r(T, L), \quad (4.43)$$

où $r(T, L) \triangleq \sum_{t=1}^T \sum_{i=1}^{\Delta} r_i(t)$.

Nous notons que pour tout bras i , $\sum_{t=1}^T r_i(t) \leq r_{\text{MBI}}(T)$. Aussi, dû au théorème 4.7.2 et $r_{\text{MBI}}(T) \leq r_{\text{OFF}}(T)$, nous avons $r_{\text{MBI}}(T) \leq m r_{\text{MBM}}(T)$. Donc, $r(T, L) \leq \Delta r_{\text{MBI}}(T) \leq m \Delta r_{\text{MBM}}(T)$.

En choisissant $j = \arg \max_{i \in \{1, 2, \dots, \Delta\}} \sum_{t=1}^T r_i(t)$ dans (4.43), nous obtenons :

$$\mathbb{E}[r_{\text{ONL}}(T)] \geq (1 - \theta) \left(r_{\text{MBI}}(T) - m \frac{\ln \Delta}{\zeta} \right) - (e - 2)\zeta \Delta r_{\text{MBI}}(T),$$

tandis qu'en choisissant $j = \Delta$ dans (4.43), nous obtenons :

$$\mathbb{E}[r_{\text{ONL}}(T)] \geq (1 - \theta) \left(r_{\text{MBM}}(T) - m \frac{\ln \Delta}{\zeta} \right) - (e - 2)\zeta m \Delta r_{\text{MBM}}(T).$$

En mettant les équations précédentes différemment, nous obtenons :

$$r_{\text{MBI}}(T) - \mathbb{E}[r_{\text{ONL}}(T)] \leq \theta(e - 1)r_{\text{MBI}}(T) + m \frac{\Delta \ln \Delta}{\theta},$$

$$r_{\text{MBM}}(T) - \mathbb{E}[r_{\text{ONL}}(T)] \leq \theta(1 + m(e - 2))r_{\text{MBM}}(T) + m \frac{\Delta \ln \Delta}{\theta}.$$

Ceci prouve la proposition.

Pour une valeur soigneusement choisie de θ , nous pouvons prouver que les deux regrets atteignent $\mathcal{O}(\sqrt{T})$. Autrement dit, les regrets augmentent d'une manière sous-linéaire en fonction du temps.

Théorème 4.7.6 *Pour un horizon temporel donné, T , si $\theta = \theta^{\text{faible}}$ (resp. $\theta = \theta^{\text{souple}}$), alors le regret faible et le regret souple sont bornés, respectivement, comme suit :*

$$\mathfrak{R}_{\text{ONL}}^w(T) \leq 2m\sqrt{(e-1)\Delta \ln \Delta} \sqrt{T}, \quad (4.44)$$

$$\mathfrak{R}_{\text{ONL}}^s(T) \leq 2m\sqrt{[1+(e-2)m]\Delta \ln \Delta} \sqrt{T}, \quad (4.45)$$

où $\theta^{\text{faible}} \triangleq \min\{1, \sqrt{\Delta \ln \Delta / ((e-1)T)}\}$ et $\theta^{\text{souple}} \triangleq \min\{1, \sqrt{\Delta \ln \Delta / ([1+(e-2)m]T)}\}$.

Démonstration : nous avons $r_{\text{MBM}}(T) \leq r_{\text{MBI}}(T) \leq mT$ puisque dans le meilleur cas MBM-OED et MBI-OED ordonnance m utilisateurs dans chaque trame. D'une part, si $\theta^{\text{faible}} = 1$ (resp. $\theta^{\text{souple}} = 1$), donc nous avons $\Delta \ln \Delta \geq T(e-1)$ et $\Delta \ln \Delta \geq T(1+(e-2)m)$. Puisque le regret faible et celui souple ne peuvent pas être plus grand que mT , nous obtenons les résultats voulus. D'autre part, si $\theta^{\text{faible}} < 1$ (resp. $\theta^{\text{souple}} < 1$), donc en utilisant la proposition 4.7.5, nous obtenons respectivement

$$\begin{aligned} \mathfrak{R}_{\text{ONL}}^w(T) &\leq \sqrt{\frac{\Delta \ln \Delta}{(e-1)T}}(e-1)mT + m\Delta \ln \Delta \sqrt{\frac{(e-1)T}{\Delta \ln \Delta}} \\ &= 2m\sqrt{\Delta \ln \Delta} \sqrt{(e-1)} \sqrt{T}, \end{aligned} \quad (4.46)$$

$$\begin{aligned} \mathfrak{R}_{\text{ONL}}^s(T) &\leq \sqrt{\frac{\Delta \ln \Delta}{[1+(e-2)m]T}}[1+(e-2)m]mT + m\Delta \ln \Delta \sqrt{\frac{[1+(e-2)m]T}{\Delta \ln \Delta}} \\ &= 2m\sqrt{\Delta \ln \Delta} \sqrt{[1+(e-2)m]} \sqrt{T}. \end{aligned} \quad (4.47)$$

Ceci prouve le théorème. ■

Les bornes supérieures montrées dans le théorème 4.7.6 sont valables lorsque T est connu à l'avance. Cependant, lorsque T n'est pas connu, nous pouvons appliquer la « technique

de devinette » de (Auer *et al.*, 2002) et obtenir les mêmes bornes supérieures qui tiennent uniformément en fonction de T .

4.7.2 Le cas de plusieurs stations de base

Dans cette section, nous étudions OAED-ON. Autrement dit, le problème concerne maintenant l'ordonnancement et l'association des utilisateurs. En utilisant la preuve du lemme 4.5.3, nous pouvons voir qu'OAED-ON est NP-difficile. De plus, en appliquant le même principe d'ALG-SFMB nous pouvons montrer que l'algorithme 15, appelé OFF-OAED, est 2-approximation pour OAED-ON dans le scénario *offline*.

Algorithme 15 L'algorithme OFF-OAED

```

1:  $\mathcal{U} \leftarrow \bigcup_t \mathcal{U}^t$ 
2:  $\mathcal{A} \leftarrow \emptyset$ 
3:  $\mathcal{B}^* \leftarrow \emptyset$ 
4: tant que  $\mathcal{B} \neq \emptyset$  et  $\mathcal{U} \neq \emptyset$  faire
5:   pour  $b$  dans  $\mathcal{B}$  faire
6:      $\mathcal{A}_b \leftarrow \text{OFF}(\mathcal{U}, \mathcal{A}_b, m, T, \mathbf{v}_b)$ 
7:   fin pour
8:    $b^* \leftarrow \arg \max \{\mathcal{A}_b\}$ 
9:    $\mathcal{B} \leftarrow \mathcal{B} \setminus \{b^*\}$ 
10:   $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{A}_{b^*}$ 
11:   $\mathcal{B}^* \leftarrow \mathcal{B}^* \cup \{b^*\}$ 
12: fin tant que
13: retourner  $\mathcal{A}_{\mathcal{B}^*}$ 

```

Dans ce qui suit, nous proposons l'algorithme compétitif ON-OAED pour résoudre OAED-ON dans le scénario *online*. ON-OAED utilise ON-OED itérativement. Pour chaque trame, ON-OAED itère les stations de base et choisit celle qui ordonnance le nombre maximum d'utilisateurs. Ensuite, il supprime cette station de base (ainsi que ses utilisateurs associés) et continue à itérer le reste des stations de base jusqu'à ce qu'aucun autre utilisateur ne puisse être ordonnancé ou que tous les utilisateurs soient ordonnancés dans la trame actuelle.

En suivant la démonstration du théorème 4.7.2, nous pouvons prouver qu'ON-OAED est un algorithme polynomial m -compétitif. En effet, le nombre de *slots* utilisés par un algorithme *offline* optimal OFF-OAED peut être écrit comme suit : $v(\text{MOFF}) \leq \sum_{b=1}^n L_{b,T+1} + v(\text{MON})$, où $v(\text{MOFF})$ représente le nombre de *slots* utilisés par OFF-OAED et $v(\text{MON})$ représente le nombre de *slots* utilisés par ON-OAED et $L_{b,T+1}$ est l'énergie restante au niveau de la station de base b après l'horizon du temps T .

Comme nous avons fait dans la démonstration du théorème 4.7.2, nous pouvons montrer qu'il existe une constante $\alpha = nA_{\max}$ telle que $0 \leq m\alpha + nA_{\max}$, où 0 (resp. α) est le nombre d'utilisateurs servis par OFF-OAED (resp. par ON-OAED). Ainsi, ON-OAED est un algorithme polynomial m -compétitif.

Remarque 4.7.7 (Une borne supérieure étroite) *Nous donnons une instance de OAED-ON pour laquelle ON-OAED atteint un ratio compétitif de m . Soit $T = 2q$ pour un entier positif q . L'arrivée d'énergie est m si t est impair et 0 sinon. Pour chaque trame t , il y a n ou mn utilisateurs, c'est-à-dire que $\mathcal{U}_t = \mathcal{U}_{\text{impair}}(t) = \{1, 2, \dots, n\}$ si t est impair et $\mathcal{U}_t = \mathcal{U}_{\text{pair}}(t) = \{1, 2, \dots, mn\}$ sinon, où $v_{b \rightarrow u}^t = m$ si t est impair (pour tout b et pour $u \in \mathcal{U}_{\text{impair}}(t)$) et $v_{b \rightarrow u}^t = 1$ (pour tout b et pour $u \in \mathcal{U}_{\text{pair}}(t)$) sinon.*

ON-OAED sert un utilisateur dans chaque trame impaire et aucun utilisateur dans chaque trame paire et ceci pour chaque station de base b . Par conséquent, au total, ON-OAED sert qn utilisateurs. Cependant, OFF-OAED ne sert aucun utilisateur dans les trames impaires et m utilisateurs dans chaque trame paire et ceci pour chaque station de base b . Au total, OFF-OAED sert mnq utilisateurs. Nous pouvons voir que le rapport, *offline-sur-online*, du nombre d'utilisateurs servis est m . Par conséquent, ON-OAED atteint le meilleur ratio compétitif possible.

4.7.3 Résultats de simulations

Nous considérons une zone géographique où les utilisateurs et les stations de base sont distribués de manière aléatoire et uniforme. Les gains des canaux sans fil entre les utilisateurs et les stations de base sont basés sur les spécifications 3GPP et sont modélisés comme dans (Zhuang *et al.*, 2016). Pour chaque trame t , nous générons $k_t \leq 2m$ utilisateurs de manière aléatoire et uniforme. L'arrivée d'énergie $E_{b,t}$ est supposée suivre une distribution de Poisson avec le paramètre λ , alors que les exigences des utilisateurs en matière de données sont choisies uniformément dans $[1, 100]$ mégabits. La fréquence de porteuse est de 2 GHz, la largeur de bande de $B = 10$ MHz et la puissance du bruit est de $\sigma^2 = -174$ dBm/Hz.

Nous commençons par présenter les résultats de simulation pour OED-ON où il existe une seule station de base qui utilise une puissance de transmission égale à 30 dBm. Le gain de canal est donné par $g_u^t = 30,6 + 36,7 \log_{10} d_u^t$, où d_u^t est la distance entre u et la station de base à la trame t . Nous comparons ON-OED à l'algorithme *offline* OFF-OED et à un algorithme de référence appelé RON-OED qui est une version randomisée de ON-OED. Autrement dit, au lieu d'utiliser toute l'énergie disponible dans chaque trame, RON-OED utilise une quantité aléatoire.

La figure 4.16 montre que si T augmente, alors ON-OED sert un nombre d'utilisateurs proche de celui d'OFF-OED. Par exemple, pour $T = 100$, nous pouvons voir que le rapport entre les utilisateurs servis par OFF-OED et ceux servis par ON-OED est au plus égal à 1,11 pour $\lambda = 2$. Ce ratio est presque cinq fois meilleur que le facteur compétitif de $m = 5$. Nous pouvons également voir que, lorsque λ augmente, les performances d'ON-OED et d'OFF-OED sont presque identiques. Par exemple, le ratio, *offline-sur-online*, du nombre d'utilisateurs servis est égal à environ 1,006 pour $\lambda = 5$. En effet, à mesure que l'énergie devient toujours disponible, le problème n'est plus de savoir comment utiliser l'énergie

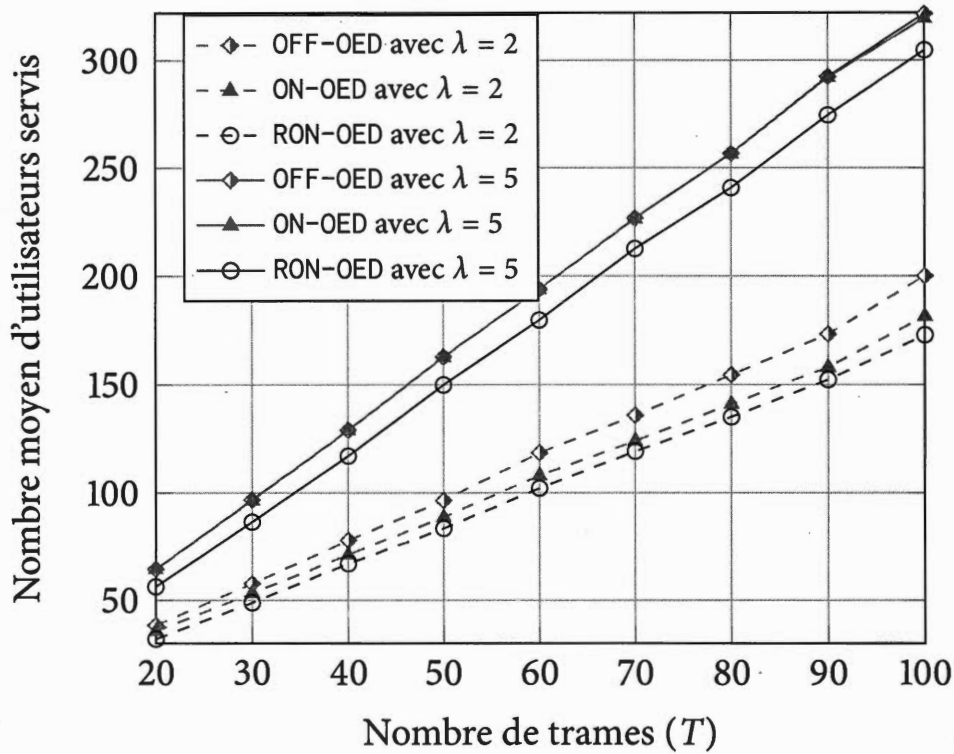


Figure 4.16 Nombre moyen d'utilisateurs servis par tous les algorithmes pour OED-ON en fonction de T pour $m = 5$.

disponible (qui est abondante) mais il est de savoir comment exploiter les trames disponibles, qui est une tâche simple. Donc, ON-OED prend presque toujours des décisions optimales. Bien que l'aspect aléatoire permette à RON-OED de faire varier les quantités d'énergie utilisées à chaque trame comme le fait OFF-OED, ON-OED donne des meilleures performances par rapport à RON-OED. Ceci est dû au fait que RON-OED ignore ses décisions passées et il ne met pas à jour ses choix.

La figure 4.17 montre qu'ON-OED est proche d'OFF-OED. Le rapport, *offline-sur-online*, de nombre d'utilisateurs servis pour $\lambda = 5$ et $m = 10$ est presque égal à 1,1, ce qui est beaucoup mieux que le ratio théorique de $m = 10$. Même si ce rapport augmente avec m , nous pouvons affirmer que pour des valeurs plus grandes de λ , ON-OED s'approche

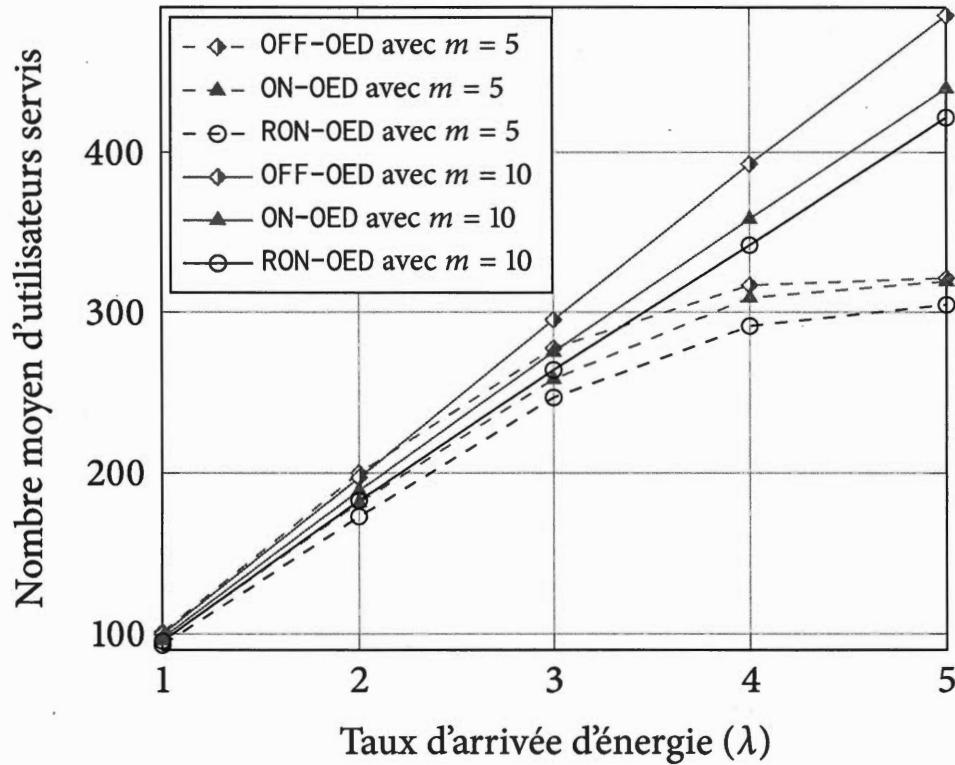


Figure 4.17 Nombre moyen d'utilisateurs servis par tous les algorithmes pour OED-ON en fonction λ pour $T = 100$.

d'OFF-OED. Nous remarquons aussi que pour des petites valeurs de λ , par exemple $\lambda = 1$, les performances de tous les algorithmes sont presque identiques, car ils ordonnancent tous peu d'utilisateurs (si jamais).

Les deux figures précédentes montrent que les performances d'ON-OED sont beaucoup mieux en simulation qu'en théorie. De plus, utiliser toute l'énergie disponible dans le cas où sa quantité est petite ou grande est presque optimal lorsque l'entrée du problème n'est pas connue a priori.

Dans les simulations suivantes, nous présentons le regret en fonction du temps ($\mathfrak{R}_O\mathfrak{T}$) d'ONL-OED, défini comme suit : $\mathfrak{R}_O\mathfrak{T} \triangleq \mathfrak{R}_{\text{ONL}}(T)/T$, où $\mathfrak{R}_{\text{ONL}}(T)$ est donné par (4.31) ou (4.33).

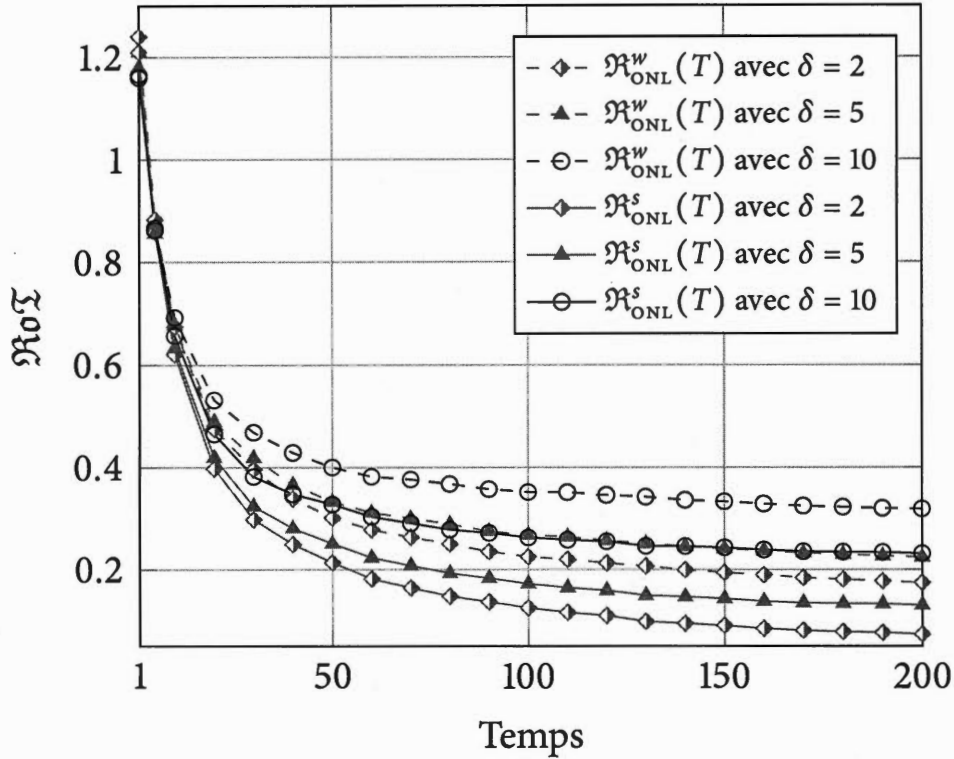


Figure 4.18 Le regret en fonction du temps pour OED-ON ($n = 5$ et $\lambda = 2$).

La figure 4.18 montre en effet que les regrets faible et souple croissent de manière sous-linéaire avec le temps, c'est-à-dire que lorsque le temps augmente, $\mathcal{R}_O\mathcal{I}$ s'approche de zéro. De plus, à mesure que δ diminue (plus d'énergie est disponible), le regret diminue car le nombre de bras disponibles augmente. En fait, quand il y a plus de bras, ON-OED peut explorer plusieurs choix et donc peut apprendre mieux. Lorsque le temps est petit, ONL-OED apprend plus rapidement lorsque δ est grand, car le nombre de bras est faible et il peut donc les explorer plus rapidement en moins de temps. Cependant, lorsque T croît, ONL-OED apprend plus vite lorsque δ est petit car le nombre de bras augmente et permet donc d'exploiter-puis-explorer de meilleurs bras avec le temps. Cependant, ONL-OED devient de plus en plus complexe et il faudra donc résoudre efficacement le problème du compromis entre performance et complexité. La figure 4.18 montre également que le

regret souple est meilleur que le regret faible.

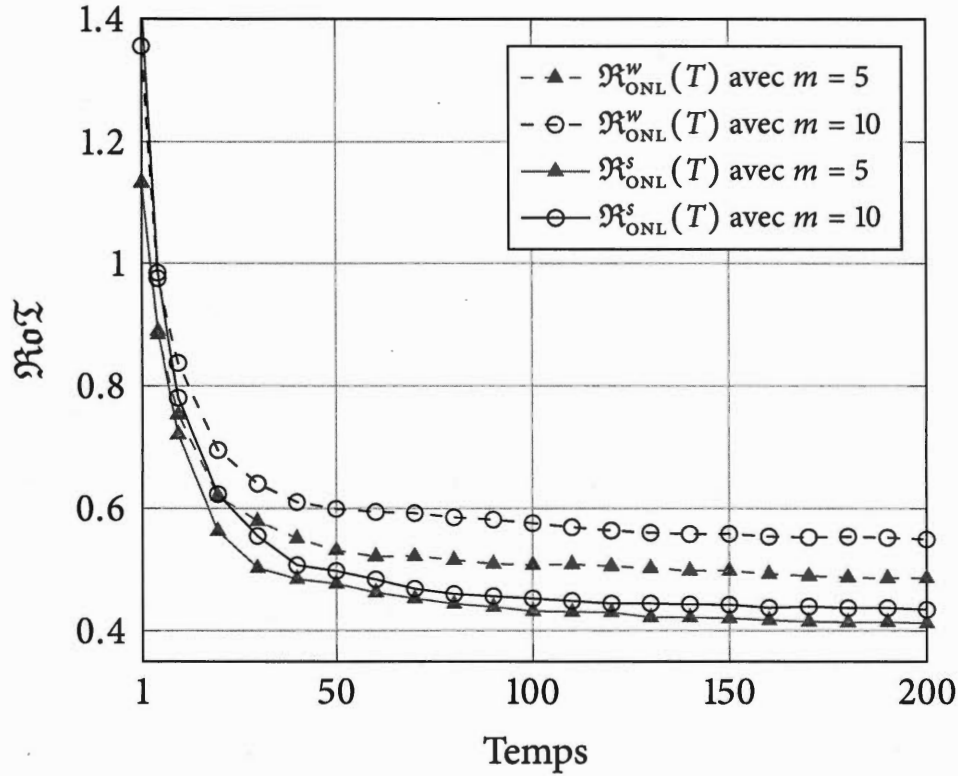


Figure 4.19 Le regret en fonction du temps pour OED-ON ($\delta = 20$ et $\lambda = 2$).

La figure 4.19 montre que, lorsque m augmente, bien que les regrets faible et souple augmentent, ils continuent de croître de manière sous-linéaire avec le temps. Ceci est cohérent avec les limites théoriques prouvées dans (4.44) et (4.45).

Dans la suite, nous considérons un réseau DCN avec plusieurs stations de base. Il existe quatre stations de base transmettant avec une puissance de 35 dBm. Les $n - 4$ stations de base restantes transmettent avec 30 dBm. Les gains des canaux sans fil entre les utilisateurs u pendant les trames t sont donnés par $30,6 + 36,7 \log_{10} d_{b \rightarrow u}^t$ pour les quatre premières stations de base et par $15,3 + 37,6 \log_{10} d_{b \rightarrow u}^t$ pour le reste des stations de base. Nous comparons ON-OAED, OFF-OAED et RON-OAED (RON-OAED utilise RON-OED

de manière itérative dans chaque station de base).

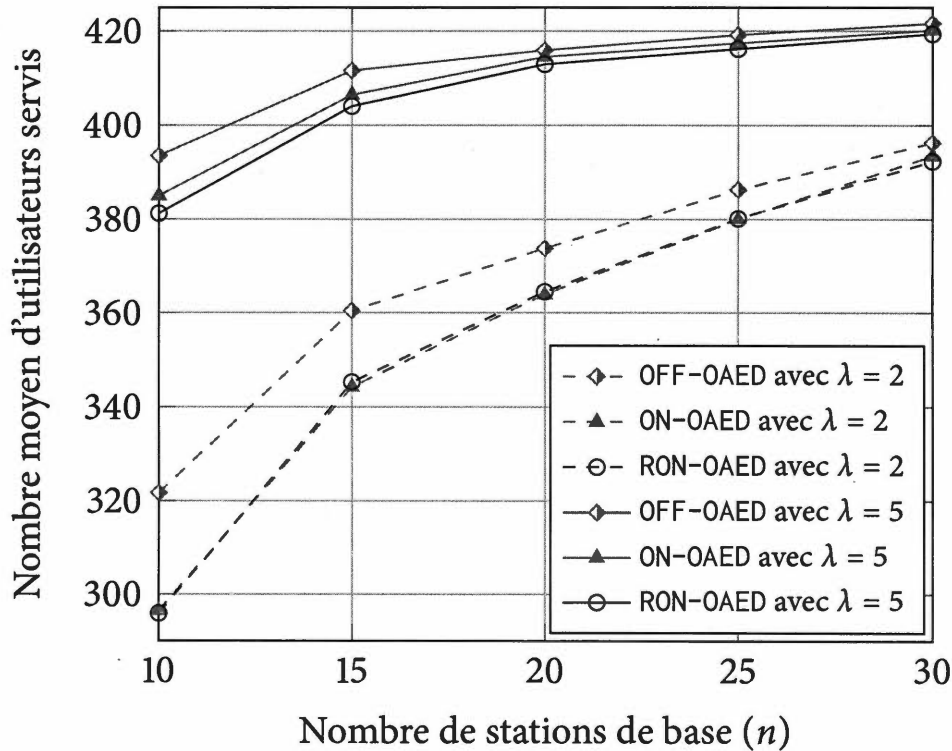


Figure 4.20 Nombre moyen d'utilisateurs servis par tous les algorithmes pour OAED-ON en fonction de n ($m = 10$ et $T = 50$).

La figure 4.20 montre que lorsque n augmente, le gap (et donc le ratio *offline-sur-online*), d'utilisateurs servis diminue, par exemple, pour $\lambda = 2$, le ratio passe de 1,1 pour $n = 10$ à 1,008 pour $n = 30$. Au fur et à mesure que de l'énergie devient disponible, ON-OAED commence à prendre des décisions quasi-optimales, en particulier pour des valeurs de n élevées. Nous concluons que la performance d'ON-OAED est meilleure en simulation que celle théorique. De plus, sa performance intéressante pour les grandes valeurs de n supporte son implémentation dans des réseaux DCNs.

4.8 Conclusions

Dans ce chapitre, nous avons étudié un problème d'allocation des ressources dans un réseau DCN avec des contraintes de dates limites et d'énergie. Ce problème, appelé PAR, consiste à associer les utilisateurs, les ordonnancer selon leurs dates d'arrivée et leurs dates limites en leur allouant les fréquences disponibles et en respectant les contraintes de récolte d'énergie. Tout d'abord, nous avons étudié un cas particulier de PAR lorsque l'énergie est toujours disponible. Nous avons étudié la NP-difficulté de ce problème et puis nous avons proposé un algorithme d'approximation. Après, nous avons montré ses performances en le comparant avec l'optimal et avec des algorithmes heuristiques. Ensuite, nous avons étudié PAR dans le cas de la récolte d'énergie mais avec des temps d'arrivée identiques. Nous avons caractérisé la NP-difficulté de PAR dans différents cas. Selon le cas nous avons proposé (i) un algorithme optimal polynomial, (ii) un algorithme d'approximation, et (iii) un algorithme heuristique. Les simulations ont montré que les performances des algorithmes proposés atteignent des valeurs proches de l'optimal. Dans un dernier lieu, nous avons étudié PAR dans le scénario *online* où son entrée n'est pas disponible au début mais arrive au fur et à mesure. Nous avons proposé des algorithmes *onlines* compétitifs ainsi qu'un algorithme d'apprentissage incrémental et nous avons montré leurs performances via des simulations.

CHAPITRE V

CONCLUSIONS ET TRAVAUX FUTURS

Ce chapitre conclut la thèse et mentionne des nouvelles pistes de recherche pour les travaux futurs.

5.1 Conclusions

Nous avons présenté tout au long de cette thèse des solutions algorithmiques au problème d'allocation des ressources dans les réseaux cellulaires denses de la prochaine génération des réseaux sans fil. Dans les réseaux cellulaires denses, plusieurs stations de base coexistent dans une seule zone géographique, ce qui permet d'augmenter l'efficacité spectrale du réseau au détriment d'augmenter les sources d'interférence. Les algorithmes de gestion d'interférence qui existent dans la littérature ne permettent pas de tirer profit de tous les avantages offerts. L'optimisation de l'allocation des ressources dont disposent les réseaux cellulaires est évidemment l'une des meilleures méthodes qui peuvent efficacement gérer l'interférence. Dans cette thèse, nous avons optimisé les ressources spatiales, énergétiques, temporelles et fréquentielles afin de maximiser le nombre d'utilisateurs servis dans les réseaux cellulaires denses de la cinquième génération. Nos objectifs étaient de caractériser les classes de complexité de ce problème et de développer des algorithmes efficaces avec des garanties de performance dans les pires cas.

Nous avons, premièrement, étudié l'optimisation de l'allocation de ressource spatiale. Cette ressource représente les stations de base et elle est considérée comme la ressource la plus importante de notre travail. Puisque la principale caractéristique des réseaux cellulaires denses est le déploiement de beaucoup de stations de base dans une seule zone géographique, donc l'association des utilisateurs aux différentes stations de base est devenue un problème crucial qui nécessite une attention particulière. Le problème considéré est la maximisation des utilisateurs associés aux stations de base sous les contraintes des rapports signal-sur-interférence-plus-bruit (SINR). Puisqu'une station de base peut associer plusieurs utilisateurs, nous avons considéré un accès multiple temporel afin de permettre aux utilisateurs d'une même station de base de la partager. Ainsi, la maximisation est faite sur un ensemble de *slots*, appelé *multi-slot association*. Nous avons étudié la NP-difficulté de ce problème ainsi que sa version *uni-slot*. Notre étude a montré que ces deux problèmes sont NP-difficiles. Nous avons proposé un algorithme qui résout MSA itérativement dans chaque *slot*. Cet algorithme garantit une borne supérieure étroite sur la valeur de la solution optimale de MSA. Ensuite, nous avons résolu MSA dans un scénario distribué où l'existence d'un point central qui collecte toutes les informations nécessaires n'est pas garantie. Nous l'avons modélisé en utilisant la théorie des jeux et nous nous sommes inspirés d'un algorithme d'apprentissage bien connu qui résout le dilemme du prisonnier. Cet algorithme est totalement distribué dans le sens où aucune communication n'est requise entre les joueurs. Nous avons montré que cet algorithme permet de converger vers un état d'équilibre maximisant le nombre d'utilisateurs associés. Des simulations Monte Carlo ont montré l'efficacité des algorithmes centralisés et distribués proposés.

Deuxièmement, nous avons étudié l'impact d'association des utilisateurs sur la consommation énergétique. En effet, puisque l'objectif est de maximiser le nombre d'utilisateurs associés, un nombre important de stations de base seront activées. La consommation énergétique sera donc élevée. Nous avons formulé le problème d'association des

utilisateurs comme un programme linéaire en nombres entiers qui a pour objectif de minimiser le nombre de stations de base à activer sous les contraintes d'associer tous les utilisateurs et de satisfaire leurs exigences de qualité de service. Ensuite, nous avons étudié la NP-difficulté de ce problème dans deux cas, notamment lorsque la qualité de service exigée est élevée ou non. Nous avons montré que même trouver une solution faisable de ce problème est NP-difficile dans les deux cas. Lorsque la qualité de service est élevée, nous avons développé une technique qui permet de simplifier le problème énormément. En effet, nous avons montré qu'il suffit de trouver seulement l'ensemble de stations de base à activer, c.-à-d. cet ensemble permet de donner facilement les associations des utilisateurs. En se basant sur cette technique, nous avons pu développer un algorithme de force brute optimal de complexité réduite—bien qu'elle reste toujours exponentielle. En se basant sur cet algorithme optimal, nous avons conçu un algorithme glouton de complexité polynomiale. D'autre part, lorsque la qualité de service est faible, le problème d'activation de stations de base est très couplé avec celui d'association des utilisateurs et donc la technique développée précédemment ne fonctionnait pas. En utilisant l'hypothèse de pire cas d'interférence, c.-à-d. toutes les stations de base interfèrent entre elles, nous avons conçu un algorithme polynomial d'approximation qui donne : (i) un facteur logarithmique si le problème est faisable et (ii) un facteur constant sinon. Des simulations Monte Carlo montrent la supériorité des solutions proposées par rapport aux solutions optimales complexes ou existantes dans la littérature.

Troisièmement, nous avons étudié le problème conjoint d'ordonnancement, d'association des utilisateurs et d'allocation de fréquences. Ce problème était une extension nécessaire des problèmes discutés précédemment. Bien que l'allocation universelle de fréquences permette d'augmenter l'efficacité spectrale, elle limite toujours les performances à cause de l'interférence sévère qui peut exister sur une seule fréquence. Par conséquent, l'allocation des ressources fréquentielles est obligatoire. De plus, les utilisateurs ont généralement des requêtes avec des exigences et des contraintes différentes, et donc ils

doivent s'associer aux stations de base pendant différentes périodes afin satisfaire leurs requêtes. Ainsi, l'ordonnancement des utilisateurs est aussi obligatoire. En outre, vue la consommation énergétique importante des stations de base, nous avons supposé que ces dernières utilisent la technologie éco-responsable de récolte d'énergie. Afin de simplifier l'étude de ce problème conjoint d'ordonnancement, d'association des utilisateurs et d'allocation de fréquences avec récolte d'énergie, nous l'avons divisé en des sous-problèmes. Nous avons commencé par le problème d'ordonnancement et d'association des utilisateurs sans récolte d'énergie. Nous avons montré que ce problème peut être résolu dans un temps polynomial à condition que les temps d'arrivée des requêtes des utilisateurs soient identiques et qu'il n'y ait une seule station de base. Dans le cas d'une arrivée arbitraire ou de plusieurs stations de base, nous avons montré qu'il s'agit d'un problème NP-difficile et nous avons développé un algorithme d'approximation simple qui garantit d'obtenir une solution avec une valeur qui n'est pas pire que de deux fois de celle de la solution optimale. Les résultats de simulations ont montré l'efficacité de cet algorithme par rapport à l'optimal et d'autres algorithmes heuristiques. Ensuite, nous avons considéré un cas plus général du problème en étudiant l'ordonnancement, l'association des utilisateurs et l'allocation de fréquences avec la récolte d'énergie. De la même façon, nous avons divisé ce problème en des sous-problèmes et nous avons analysé la NP-difficulté de chacun. Lorsque le problème est dans la classe de complexité P nous avons développé un algorithme polynomial simple. Lorsqu'il s'agit d'un problème NP-difficile, d'une part nous avons proposé un algorithme d'approximation pour l'ordonnancement, l'association des utilisateurs avec récolte d'énergie et d'autre part nous avons développé un algorithme heuristique pour le problème général. Finalement, nous avons étudié le problème d'ordonnancement et d'association des utilisateurs avec récolte d'énergie dans un scénario *online*. Ce scénario signifie que l'entrée de ce problème n'est pas disponible en entier mais qu'elle arrive comme un flux de données au fur et à mesure que le problème est résolu. Le défi ici est qu'un algorithme pour résoudre ce problème doit prendre des décisions

irrévocables à chaque fois qu'une nouvelle partie de l'entrée est arrivée. Nous avons analysé ce problème en utilisant deux approches : l'analyse compétitive et l'apprentissage *online*. Ensuite, nous avons proposé dans la première approche des algorithmes compétitifs (c.-à-d., des algorithmes *onlines* d'approximation) et dans la deuxième approche un algorithme d'apprentissage incrémental qui minimise le regret. Des simulations Monte Carlo ont montré les avantages des algorithmes *onlines* proposés en les comparant avec des algorithmes *offline* et heuristiques classiques.

Nous avons étudié dans cette thèse plusieurs problèmes d'optimisation de l'allocation des ressources dans les futures générations des réseaux cellulaires d'un point de vue principalement théorique. Notre étude a montré que la majorité de ces problèmes sont très difficiles à résoudre dans un temps raisonnable. Ceci est dû principalement à la nature couplée de ces problèmes et à leurs contraintes en conflit. Pour remédier à ces difficultés, nous avons suivi une approche bien connue en ingénierie soit celle de « *chercher des solutions presque optimales rapidement* ». Puisque la majorité des travaux de recherche ne fournissent pas des garanties sur ces solutions optimales, cette thèse a comblé ces lacunes en fournissant des telles garanties.

Le travail réalisé dans cette thèse est valide pour les problèmes d'allocation des ressources dans les futurs réseaux sans fil en général ainsi que ceux de la cinquième génération des réseaux cellulaires. En effet, les réseaux sans fil du futur (au-delà du cinquième génération) se caractérisent aussi par leur densité, p. ex. l'internet des objets. Ainsi, à titre d'exemple, les algorithmes proposés dans cette thèse permettent d'optimiser les ressources dans l'internet des objets afin de garantir des exigences des applications temps réel.

5.2 Travaux futurs

Les travaux réalisés dans cette thèse ouvrent plusieurs pistes de recherche pour le futur. Parmi ces pistes nous citons les suivantes.

- Cette thèse considère souvent la fonction objectif qui représente le nombre d'utilisateurs associés. L'étude d'autres fonctions objectif est évidemment une piste de recherche importante et ouverte surtout pour développer des algorithmes d'approximation et compétitifs. Évidemment, l'extension des algorithmes proposés dans cette thèse pour qu'ils optimisent d'autres fonctions objectif est également importante. Par exemple, optimiser l'âge de l'information des paquets avec des dates limites dans un réseau sans fil ou optimiser le délai de transmission pour des applications temps réel.
- Cette thèse considère seulement une technologie clé de la cinquième génération soit celle des réseaux cellulaires denses. La considération de nouveaux systèmes qui prennent en compte d'autres technologies comme la communication millimétrique ou la communication à antennes multiples à grande échelle est également une piste de recherche importante. En outre, les algorithmes proposés dans cette thèse peuvent inspirer des solutions qui fonctionnent dans ces systèmes.
- Dans le chapitre 4, nous avons proposé deux algorithmes m -compétitifs pour résoudre le problème *online* d'ordonnancement et d'association des utilisateurs. L'étude de l'optimalité du facteur m est un problème ouvert, autrement dit, existe-il un algorithme $\mathcal{O}(1)$ -compétitif pour résoudre ce problème?
- Dans le chapitre 4, lorsque nous avons étudié le problème d'ordonnancement, d'association des utilisateurs et d'allocation des fréquences avec récolte d'énergie, nous avons fourni seulement un algorithme heuristique à cause de la nature très couplée de ce problème. Développer des algorithmes d'approximation pour ce problème est important et reste un problème ouvert.

- Dans le chapitre 4, lorsque nous avons étudié le problème *online* d'ordonnement et d'association des utilisateurs avec récolte d'énergie, nous avons considéré seulement une approche centralisée. Étudier le scénario *online* distribué est également important. Aussi, développer des algorithmes distribués pour le problème d'activation des stations de base étudié dans le chapitre 3 reste un problème ouvert.
- Nous pouvons aussi investiguer l'allocation des ressources dans les futurs réseaux sans fil tel que l'internet des objets afin de garantir une communication temps réel efficace.

PUBLICATIONS LIÉES À CE PROJET

Mlika, Z., Driouch, E. et Ajib, W. (2017a). A Fully Distributed Algorithm For User-Base Station Association in HetNets. *Computer Communications*, 105, 66–78.

Mlika, Z., Driouch, E. et Ajib, W. (2017b). A Simple Approximation Algorithm for Base Station Association in HetNets. Dans *Proc. IEEE Int. Conf. on Commun. (ICC 2017)*, 1–6.

Mlika, Z., Driouch, E. et Ajib, W. (2017c). Base Station Operation and User Association in HetNets : Complexity and Heuristic Algorithms. Dans *Proc. IEEE Global Commun. Conf. (GLOBECOM 2017)*, 1–6.

Mlika, Z., Driouch, E. et Ajib, W. (2017d). User Association and Scheduling With Hard Deadlines in Heterogeneous Cellular Networks. *IEEE Commun. Lett.*, 21(12), 2698–2701.

Mlika, Z., Driouch, E. et Ajib, W. (2018b). Energy-Efficient Base Station Operation and Association in HetNets : Complexity and Algorithms. *IEEE Trans. Wireless Commun.*, 17(4), 2690–2702.

Mlika, Z., Driouch, E. et Ajib, W. (2018c). Resource Allocation in Green Dense Cellular Networks : Complexity and Algorithms. Soumis le 09 novembre 2018 à *IEEE Trans. Commun.*

Mlika, Z., Driouch, E. et Ajib, W. (2018d). User Association Under SINR Constraints in HetNets : Upper Bound and NP-Hardness. *IEEE Commun. Lett.*, 22(8), 1672–1675.

Mlika, Z., Driouch, E. et Ajib, W. (2018e). User Scheduling with Deadlines and Energy Harvesting Base Station. Accepté dans *Proc. IEEE Global Commun. Conf. (GLOBECOM 2018)*.

Mlika, Z., Driouch, E. et Ajib, W. (2019). Deadline Scheduling and Association in Energy Harvesting Networks : Competitive and Learning Algorithms. Soumis à *ACM Mobihoc 2019*.

RÉFÉRENCES

- 3GPP (2010). *Further Advancements for E-UTRA Physical Layer Aspects (Release 9)*. Rapport technique 3GPP TR 36.814, 3rd Generation Partnership Project (3GPP). Version 9.0.0.
- 3GPP (2013). *Technical specification group radio access network; Small cellenhancements for E-UTRA and E-UTRAN*. Tr 36.872, Version 12.1.0.
- 3GPP (2017). *User Equipment (UE) Radio Transmission and Reception; Part 1 : Range 1 Standalone*. Technical Specification (TS) 38.101-1, 3rd Generation Partnership Project (3GPP). Version 1.0.0
- Agussurja, L. et Lau, H. C. (2009). The price of stability in selfish scheduling games. *Web Intelli. and Agent Sys.*, 7(4), 321–332.
- Ali, M. S., Coucheney, P. et Coupechoux, M. (2016). Load Balancing in Heterogeneous Networks Based on Distributed Learning in Near-Potential Games. *IEEE Trans. Wireless Commun.*, 15(7), 5046–5059.
- Andrews, J., Buzzi, S., Choi, W., Hanly, S., Lozano, A., Soong, A. et Zhang, J. (2014). What Will 5G Be? *IEEE J. Sel. Areas Commun.*, 32(6), 1065–1082.
- Andrews, J. G., Claussen, H., Dohler, M., Rangan, S. et Reed, M. C. (2012). Femtocells : Past, present, and future. *IEEE J. Sel. Areas Commun.*, 30(3), 497–508.
- Andrews, M. et Dinitz, M. (2009). Maximizing Capacity in Arbitrary Wireless Networks in the SINR Model : Complexity and Game Theory. Dans *Proc. IEEE Int. Conf. on Computer Communications (INFOCOM)*, 1332–1340., Rio de Janeiro, Brazil.
- Ao, W. C. et Psounis, K. (2017). Approximation Algorithms for Online User Association in Multi-Tier Multi-Cell Mobile Networks. *IEEE Trans. Netw.*, 25(4), 2361–2374.
- Arora, S. et Barak, B. (2009). *Computational Complexity : A Modern Approach* (1st éd.). New York, NY, USA : Cambridge University Press.
- Ásgeirsson, E. I. et Mitra, P. (2011). On a game theoretic approach to capacity maximization in wireless networks. Dans *Proc. IEEE Int. Conf. on Computer Communications (INFOCOM)*, 3029–3037, Shanghai, China.

- Auer, P., Cesa-Bianchi, N., Freund, Y. et Schapire, R. (2002). The Nonstochastic Multiarmed Bandit Problem. *SIAM Journal on Computing*, 32(1), 48–77.
- Bhushan, N., Li, J., Malladi, D., Gilmore, R., Brenner, D., Damnjanovic, A., Sukhavasi, R., Patel, C. et Geirhofer, S. (2014). Network Densification : The Dominant Theme For Wireless Evolution Into 5G. *IEEE Commun. Mag.*, 52(2), 82–89.
- Borodin, A. et El-Yaniv, R. (1998). *Online Computation and Competitive Analysis*. New York, NY, USA : Cambridge University Press.
- Buchbinder, N., Lewin-Eytan, L., Menache, I., Naor, J. et Orda, A. (2012). Dynamic Power Allocation Under Arbitrary Varying Channels—An Online Approach. *IEEE/ACM Trans. Netw.*, 20(2), 477–487.
<http://dx.doi.org/10.1109/TNET.2011.2170092>
- Buzzi, S., I, C., Klein, T. E., Poor, H. V., Yang, C. et Zappone, A. (2016a). A Survey of Energy-Efficient Techniques for 5G Networks and Challenges Ahead. *IEEE J. Sel. Areas Commun.*, 34(4), 697–709.
- Buzzi, S., I, C. L., Klein, T. E., Poor, H. V., Yang, C. et Zappone, A. (2016b). A Survey of Energy-Efficient Techniques for 5G Networks and Challenges Ahead. *IEEE J. Sel. Areas Commun.*, 34(4), 697–709.
- Chakrabarti, D., Kumar, R., Radlinski, F. et Upfal, E. (2008). Mortal Multi-armed Bandits. Dans *NIPS*, 273–280.
- Chandrasekhar, V., Andrews, J. et Gatherer, A. (2008). Femtocell networks : a survey. *IEEE Commun. Mag.*, 46(9), 59–67.
- Chekuri, C. et Khanna, S. (2005). A Polynomial Time Approximation Scheme for the Multiple Knapsack Problem. *SIAM Journal on Computing*, 35(3), 713–728.
- Chen, C. H. et Feng, K. T. (2017). Statistical Distance Estimation Algorithms With RSS Measurements for Indoor LTE-A Networks. *IEEE Trans. Veh. Technol.*, 66(2), 1709–1722.
- Cisco (2017). Cisco visual networking index : Global mobile data traffic forecast update, 2016-2021. <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf>. Consulté le : 2018-08-10.
- Crescenzi, P. (1997). A Short Guide to Approximation-Preserving Reductions. Dans

- Proc. IEEE Conf. of Computational Complexity (CCC 1997)*, 262–273.
- Cygan, M., Jeż, Ł. et Sgall, J. (2016). Online knapsack revisited. *Theory of Computing Systems*, 58(1), 153–190.
- Damnjanovic, A., Montojo, J., Wei, Y., Ji, T., Luo, T., Vajapeyam, M., Yoo, T., Song, O. et Malladi, D. (2011). A Survey On 3GPP Heterogeneous Networks. *IEEE Wireless Commun.*, 18(3), 10–21.
- Deshmukh, A. et Vaze, R. (2016). Online Energy-Efficient Packet Scheduling for a Common Deadline With and Without Energy Harvesting. *IEEE J. Sel. Areas Commun.*, 34(12), 3661–3674.
- Dinitz, M. (2010). Distributed Algorithms for Approximating Wireless Network Capacity. Dans *Proc. IEEE Int. Conf. on Computer Communications (INFOCOM)*, 1–9, San Diego, CA, USA.
- Elsawy, H., Hossain, E. et Kim, D. I. (2013). HetNets with Cognitive Small Cells : User Offloading and Distributed Channel Access Techniques. *IEEE Commun. Mag.*, 51(6), 28–36.
- Fehske, A., Viering, I., Voigt, J., Sartori, C., Redana, S. et Fettweis, G. (2014). Small-Cell Self-Organizing Wireless Networks. *Proc. IEEE*, 102(3), 334–350.
- Feng, D., Jiang, C., Lim, G., Cimini, L. J., Feng, G. et Li, G. Y. (2013). A Survey of Energy-Efficient Wireless Communications. *IEEE Commun. Surveys Tuts.*, 15(1), 167–178.
- Feng, M., Mao, S. et Jiang, T. (2016). BOOST : Base Station ON-OFF Switching Strategy for Energy-Efficient Massive MIMO HetNets. Dans *Proc. IEEE 35th Int. Conf. on Computer Commun. (INFOCOM)*, 1–9.
- Fudenberg, D. et Tirole, J. (1991). *Game Theory*. Cambridge, MA : MIT Press.
- Garey, M. R. et Johnson, D. S. (1979). *Computers and Intractability : A Guide to the Theory of NP-Completeness*. New York, NY, USA : W. H. Freeman & Co.
- Ghaffar, R. et Knopp, R. (2010). Fractional frequency reuse and interference suppression for OFDMA networks. Dans *Proc. Int. Symposium on Modeling and Optim. in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 273–277.
- GLPK (2006). GLPK (GNU linear programming kit).
<http://www.gnu.org/software/glpk/>.
- Goldsmith, A. (2005). *Wireless Communications*. New York, NY, USA : Cambridge University Press.

- Goussevskaja, O., Halldórsson, M. M. et Wattenhofer, R. (2014). Algorithms for Wireless Capacity. *IEEE/ACM Trans. Netw.*, 22(3), 745–755.
- Goussevskaja, O., Wattenhofer, R., Halldorsson, M. M. et Welzl, E. (2009). Capacity of Arbitrary Wireless Networks. Dans *Proc. IEEE Int. Conf. on Computer Commun. (INFOCOM 2009)*, 1872–1880.
- Halldórsson, M. M. et Mitra, P. (2011). Wireless Capacity with Oblivious Power in General Metrics. Dans *Proc. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 1538–1548.
- Hanly, S. V., Liu, C. et Whiting, P. (2015). Capacity and Stable Scheduling in Heterogeneous Wireless Networks. *IEEE J. Sel. Areas Commun.*, 33(6), 1266–1279.
- Hobfeld, T., Skorin-Kapov, L., Heegaard, P. E. et Varela, M. (2017). Definition of QoE Fairness in Shared Systems. *IEEE Commun. Lett.*, 21(1), 184–187.
- Hossain, E., Le, L. B. et Niyato, D. (2013). *Radio Resource Management in Multi-Tier Cellular Wireless Networks*. John Wiley & Sons, Inc.
- Huang, C., Zhang, J., Poor, H. V. et Cui, S. (2016). Delay-Energy Tradeoff in Multicast Scheduling for Green Cellular Systems. *IEEE J. Sel. Areas Commun.*, 34(5), 1235–1249.
- IBM (2010). IBM ILOG CPLEX Optimizer. Récupéré de <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>
- Jain, R. K., Chiu, D.-M. W. et Hawe, W. R. (1984). *A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems*. Rapport technique, Digital Equipment Corporation.
- Koutsoupias, E. et Papadimitriou, C. (1999). Worst-case equilibria. Dans *Proc. ACM Annual Conf. on Theoretical Aspects of Computer Science (STACS)*, 404–413.
- Krishnasamy, S. et Shakkottai, S. (2015). Spectrum Sharing and Scheduling in D2D-enabled Dense Cellular Networks. Dans *Proc. Int. Symp. on Modeling and Optim. in Mobile, Ad Hoc, and Wireless Netw. (WiOpt)*, 307–314.
- Lau, V. K. N. (2005). Optimal Downlink Space-Time Scheduling Design with Convex Utility Functions-Multiple-Antenna Systems with Orthogonal Spatial Multiplexing. *IEEE Trans. Veh. Technol.*, 54(4), 1322–1333.
- Lee, G., Saad, W., Bennis, M., Mehbodniya, A. et Adachi, F. (2017). Online Ski Rental for ON/OFF Scheduling of Energy Harvesting Base Stations. *IEEE Trans. Wireless Commun.*, 16(5), 2976–2990.

- Li, Y., Sheng, M., Sun, Y. et Shi, Y. (2016). Joint Optimization of BS Operation, User Association, Subcarrier Assignment, and Power Allocation for Energy-Efficient HetNets. *IEEE J. Sel. Areas Commun.*, 34(12), 3339–3353.
- Lin, X., Ganti, R. K., Fleming, P. J. et Andrews, J. G. (2013). Towards Understanding the Fundamentals of Mobility in Cellular Networks. *IEEE Trans. Wireless Commun.*, 12(4), 1686–1698.
- Lin, Y., Bao, W., Yu, W. et Liang, B. (2015). Optimizing User Association and Spectrum Allocation in HetNets : A Utility Perspective. *IEEE J. Sel. Areas Commun.*, 33(6), 1025–1039.
- Lin, Y., Wang, Y., Li, C., Huang, Y. et Yang, L. (2016). Energy Efficient Joint User Association and Power Allocation Design in Massive MIMO Empowered Dense HetNets. Dans *Proc. IEEE 84th Veh. Technol. Conf. (VTC-Fall)*, 1–6.
- Liu, D., Wang, L., Chen, Y., El Kashlan, M., Wong, K. K., Schober, R. et Hanzo, L. (2016). User Association in 5G Networks : A Survey and an Outlook. *IEEE Commun. Surveys Tuts.*, 18(2), 1018–1044.
- Liu, G., Hou, X., Jin, J., Wang, F., Wang, Q., Hao, Y., Huang, Y., Wang, X., Xiao, X. et Deng, A. (2017). 3-D-MIMO With Massive Antennas Paves the Way to 5G Enhanced Mobile Broadband : From System Design to Field Trials. *IEEE J. Sel. Areas Commun.*, 35(6), 1222–1233.
- Lopez-Perez, D., Chu, X., Vasilakos, A. et Claussen, H. (2014). Power Minimization Based Resource Allocation for Interference Mitigation in OFDMA Femtocell Networks. *IEEE J. Sel. Areas Commun.*, 32(2), 333–344.
- Lopez-Perez, D., Guvenc, I., de la Roche, G., Kountouris, M., Quek, T. Q. S. et Zhang, J. (2011). Enhanced Inter-cell Interference Coordination Challenges in Heterogeneous Networks. *IEEE Wireless Commun.*, 18(3), 22–30.
- Lubin, M. et Dunning, I. (2015). Computing in Operations Research Using Julia. *INFORMS Journal on Computing*, 27(2), 238–248.
- McKelvey, R. D., McLennan, A. M. et Turocy, T. L. (2014). *Gambit : Software Tools for Game Theory*. Version 14.1.0. Available at <http://www.gambit-project.org/>.
- Mertikopoulos, P. et Belmega, E. V. (2016). Learning to Be Green : Robust Energy Efficiency Maximization in Dynamic MIMO-OFDM Systems. *IEEE J. Sel. Areas Commun.*, 34(4), 743–757.
- Mitchell, S., O’Sullivan, M. et Dunning, I. (2011). *PuLP : A Linear Programming Toolkit for Python*. Rapport technique, The University of Auckland, Department of

Engineering Science.

Mlika, Z., Driouch, E. et Ajib, W. (2017a). A Fully Distributed Algorithm For User-Base Station Association in HetNets. *Computer Communications*, 105, 66 – 78.

Mlika, Z., Driouch, E. et Ajib, W. (2017b). A Simple Approximation Algorithm for Base Station Association in HetNets. Dans *Proc. IEEE Int. Conf. on Commun. (ICC 2017)*, 1–6.

Mlika, Z., Driouch, E. et Ajib, W. (2017c). Base station operation and user association in hetnets : Complexity and heuristic algorithms. Dans *Proc. IEEE Global Commun. Conf. (GLOBECOM 2017)*, 1–6.

Mlika, Z., Driouch, E. et Ajib, W. (2017d). User Association and Scheduling With Hard Deadlines in Heterogeneous Cellular Networks. *IEEE Commun. Lett.*, 21(12), 2698–2701.

Mlika, Z., Driouch, E. et Ajib, W. (2018a). Energy-Efficient Base Station Operation and Association in HetNets : Complexity and Algorithms. *IEEE Trans. Wireless Commun.*, 17(4), 2690–2702.

Mlika, Z., Driouch, E. et Ajib, W. (2018b). Resource Allocation in Green Dense Cellular Networks : Complexity and Algorithms. *IEEE Trans. Wireless Commun.* soumis.

Mlika, Z., Driouch, E. et Ajib, W. (2018c). User Association Under SINR Constraints in HetNets : Upper Bound and NP-Hardness. *IEEE Commun. Lett.*, 22(8), 1672–1675.

Mlika, Z., Driouch, E. et Ajib, W. (2018d). User Scheduling with Deadlines and Energy Harvesting Base Station. Dans *Proc. IEEE Global Commun. Conf. (GLOBECOM 2018)*. accepté.

Mlika, Z., Driouch, E. et Ajib, W. (2019). Deadline Scheduling and Association in Energy Harvesting Networks : Competitive and Learning Algorithms. Dans *IEEE Trans. Wireless Commun.* en préparation.

Mlika, Z., Goonewardena, M., Ajib, W. et Elbiaze, H. (2017e). UserBase Station Association in HetSNets : Complexity and Efficient Algorithms. *IEEE Trans. Veh. Technol.*, 66(2), 1484–1495.

Nguyen, V. M. et Kountouris, M. (2017). Performance Limits of Network Densification. *IEEE J. Sel. Areas Commun.*, 35(6), 1294–1308.

Nowak, M. et Sigmund, K. (1993). A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364(6432), 56–58.

- Pinedo, M. L. (2016). *Scheduling : Theory, Algorithms, and Systems* (5th éd.). Springer Publishing Company, Incorporated.
- Pollakis, E., Cavalcante, R. L. G. et Stanczak, S. (2012). Base Station Selection for Energy-Efficient Network Operation with the Majorization-Minimization Algorithm. Dans *Proc. IEEE 13th Int. Workshop on Signal Process. Advances in Wireless Commun. (SPAWC)*, 219–223.
- Quek, T. Q. S., de la Roche, G., Gven, I. et Kountouris, M. (2013). *Small Cell Networks : Deployment, PHY Techniques, and Resource Management*. New York, NY, USA : Cambridge University Press.
- Rao, J. B. et Fapojuwo, A. O. (2014). A Survey of Energy-Efficient Resource Management Techniques for Multicell Cellular Networks. *IEEE Commun. Surveys Tuts.*, 16(1), 154–180.
- Gurobi Optimization Inc. (2016). Gurobi Optimizer Reference Manual. Récupéré de <http://www.gurobi.com>
- Schrijver, A. (1986). *Theory of Linear and Integer Programming*. New York, NY, USA : John Wiley & Sons, Inc.
- Shan, F., Luo, J., Wu, W., Li, M. et Shen, X. (2015). Discrete Rate Scheduling for Packets With Individual Deadlines in Energy-Harvesting Systems. *IEEE J. Sel. Areas Commun.*, 33(3), 438–451.
- Shen, C., Tekin, C. et van der Schaar, M. (2016). A Non-Stochastic Learning Approach to Energy Efficient Mobility Management. *IEEE J. Sel. Areas Commun.*, 34(12), 3854–3868.
- Shen, K. et Yu, W. (2014). Distributed Pricing-Based User Association for Downlink Heterogeneous Cellular Networks. *IEEE J. Sel. Areas Commun.*, 32(6), 1100–1113.
- Sohn, I. et Lee, S. (2016). Distributed Load Balancing via Message-Passing for Heterogeneous Cellular Networks. *IEEE Trans. Veh. Technol.*, 65(11), 9287–9298.
- Son, K., Kim, H., Yi, Y. et Krishnamachari, B. (2011). Base Station Operation and User Association Mechanisms for Energy-Delay Tradeoffs in Green Cellular Networks. *IEEE J. Sel. Areas Commun.*, 29(8), 1525–1536.
- Tam, H. H. M., Tuan, H. D., Ngo, D. T., Duong, T. Q. et Poor, H. V. (2017). Joint Load Balancing and Interference Management for Small-Cell Heterogeneous Networks With Limited Backhaul Capacity. *IEEE Trans. Wireless Commun.*, 16(2), 872–884.
- TEKTELIC Communications, I. (2014). Base station density must increase.

<https://www.maketecheasier.com/whats-5g-going-look-like/>. Consulté le : 2018-11-12.

Thielen, C., Tiedemann, M. et Westphal, S. (2016). The Online Knapsack Problem with Incremental Capacity. *Mathematical Methods of Operations Research*, 83(2), 207–242.

Vaze, R. (2013). Competitive Ratio Analysis of Online Algorithms to Minimize Packet Transmission Time in Energy Harvesting Communication System. Dans *IEEE INFOCOM*, 115–123.

Vaze, R., Garg, R. et Pathak, N. (2014). Dynamic power allocation for maximizing throughput in energy-harvesting communication system. *IEEE/ACM Trans. Netw.*, 22(5), 1621–1630.

Vazirani, V. V. (2010). *Approximation Algorithms*. Springer Publishing Company, Incorporated.

Wang, N., Hossain, E. et Bhargava, V. K. (2016). Joint Downlink Cell Association and Bandwidth Allocation for Wireless Backhauling in Two-Tier HetNets With Large-Scale Antenna Arrays. *IEEE Trans. Wireless Commun.*, 15(5), 3251–3268.

Wang, Y. C. et Chen, S. T. (2017). Delay-Aware ABS Adjustment to Support QoS for Real-Time Traffic in LTE-A HetNet. *IEEE Wireless Commun. Lett.*, 6(5), 590–593.

Williamson, D. P. et Shmoys, D. B. (2011). *The Design of Approximation Algorithms* (1st éd.). New York, NY, USA : Cambridge University Press.

Xu, X., Sun, Z., Dai, X., Svensson, T. et Tao, X. (2017). Modeling and Analyzing the Cross-Tier Handover in Heterogeneous Networks. *IEEE Trans. Wireless Commun.*, 16(12), 7859–7869.

Ye, G., Zhang, H., Liu, H., Cheng, J. et Leung, V. C. M. (2016a). Energy Efficient Joint User Association and Power Allocation in a Two-Tier Heterogeneous Network. Dans *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 1–5.

Ye, Q., Bursalioglu, O. Y., Papadopoulos, H. C., Caramanis, C. et Andrews, J. G. (2016b). User Association and Interference Management in Massive MIMO HetNets. *IEEE Trans. Commun.*, 64(5), 2049–2065.

Zhao, J., Liu, Y., Chai, K. K., Nallanathan, A., Chen, Y. et Han, Z. (2017). Spectrum Allocation and Power Control for Non-Orthogonal Multiple Access in HetNets. *IEEE Trans. Wireless Commun.*, 16(9), 5825–5837.

Zhou, H., Mao, S. et Agrawal, P. (2015). Approximation Algorithms for Cell Association and Scheduling in Femtocell Networks. *IEEE Trans. Emerg. Topics*

Comput., 3(3), 432–443.

Zhuang, B., Guo, D. et Honig, M. L. (2016). Energy-Efficient Cell Activation, User Association, and Spectrum Allocation in Heterogeneous Networks. *IEEE J. Sel. Areas Commun.*, 34(4), 823–831.

Zhuang, B., Guo, D., Wei, E. et Honig, M. L. (2017a). Scalable Spectrum Allocation and User Association in Networks With Many Small Cells. *IEEE Trans. Commun.*, 65(7), 2931–2942.

Zhuang, B., Guo, D., Wei, E. et Honig, M. L. (2017b). Scalable Spectrum Allocation and User Association in Networks With Many Small Cells. *IEEE Trans. Commun.*, 65(7), 2931–2942.

Zhuang, B., Guo, D., Wei, E. et Honig, M. L. (2018). Large-Scale Spectrum Allocation for Cellular Networks via Sparse Optimization. *IEEE Trans. Signal Process.*, 66(20), 5470–5483.