



Algorithmes d'allocation de ressources en OFDMA : cas de la voie descendante d'une cellule radio isolée

Carle Lengoumbi
Philippe Martins
Philippe Godlewski

2006D007

octobre 2006

Département Informatique et Réseaux
Groupe Réseaux, Mobilité et Sécurité



Carle Lengoumbi

Algorithmes d'allocation de ressources en OFDMA :
Cas de la voie descendante d'une cellule radio isolée

Directeur de thèse : Philippe Martins
Co-directeur de thèse: Philippe Godlewski

Octobre 2006

Table des matières

Table des matières	2
Liste des tableaux	3
Table des illustrations	3
Notations	4
Acronymes	4
Résumé	5
1. Introduction	6
1.1. Contexte de l'OFDMA	6
1.2. Problèmes d'optimisation avec contrainte en OFDMA	6
2. Algorithmes d'allocation des ressources	9
2.1. Décomposition par « étapes » dans les problèmes MA	9
2.2. Décomposition par « étapes » dans les problèmes RA	11
2.3. BARE et RPO, deux contributions au problème RA	12
3. Tâche 1 : Allocation de largeur de bande	13
3.1. La tâche 1 dans la littérature	13
3.2. Une contribution pour la tâche 1 en RA : le BARE	15
4. Tâche 2 : Affectation des sous porteuses	18
4.1. Algorithme Hongrois	18
4.2. Algorithmes heuristiques en MA	19
4.3. Algorithmes heuristiques en RA	23
4.4. Un nouvel algorithme : le RPO	24
5. Simulation et résultats	27
5.1. Motivations	27
5.2. Modèle canal et Paramètres du système	27
5.3. Débit par sous porteuse	28
5.4. Facteur d'équité en débit entre les utilisateurs	31
5.5. Probabilité d'échec de la tâche 2	31
5.6. Complexité des algorithmes de la tâche 2	33
6. Conclusion	34
Annexe A : Puissance démission et <i>SNR gap</i>	35
Annexe B : L'algorithme Hongrois	36
Annexe C : Le <i>waterfilling</i>	38
Références	41

Liste des tableaux

Tableau 1: Fonctions "puissance-débit"	7
Tableau 2: Notations de l'algorithme RPO	25
Tableau 3: Paramètres de simulation.....	28

Table des illustrations

Figure 1: Fonction "débit-puissance" considérée dans [Kivanc&_03]	8
Figure 2: Décomposition 1 en MA ([WongMurch_99], [Inyoung&_01])	10
Figure 3: Décomposition 2 en MA ([WongTsui_99], [PietrykJan_02], [Ergen_03], [Chen&_04])	10
Figure 4: Décomposition en RA ([Kivanc&_00], [Kivanc&_03], [YinLiu_00], [Wong&_04], [Anas_04]).....	11
Figure 5: Principe du <i>waterfilling</i>	11
Figure 6: Phase d'allocation de sous porteuses ([Kivanc&_00], [Kivanc&_03], [YinLiu_00], [Gross&_03]).....	12
Figure 7: Echange de modulation entre 2 sous porteuses.....	21
Figure 8: Comparaison de BABS et BARE (débit moyen par sous porteuses)	29
Figure 9: Débit moyen par sous porteuse	30
Figure 10: Facteur d'équité, $F = \min_u r_u / \max_u r_u$	31
Figure 11: Débit minimal commun à la fin de la tâche 1 (exprimé en nombre de bits par symboles OFDM) en fonction de la puissance par sous porteuses en mW	32
Figure 12: Probabilité d'échec	32
Figure 13: Durée d'exécution des algorithmes (tâche 2) pour 4 utilisateurs	33
Figure 14: Niveau d'eau dans le <i>waterfilling</i>	40

Notations

U	:	nombre total d'utilisateurs
u, v	:	indices sur l'utilisateur u
W	:	bande de fréquence totale
Δf	:	écart entre 2 sous porteuses consécutives
N_0	:	bruit blanc gaussien (-174dBm/Hz)
σ_{sh}	:	écart type du fading log normal
N	:	nombre total de sous porteuses (utiles)
n, m	:	indices sur les sous porteuses
N_u	:	nombre total de sous porteuses attribuées à u
$\mathcal{Q}_u$	:	ensemble des sous porteuses affectées à l'utilisateur u
$P_{T,max}$	:	budget de puissance total
P_u	:	budget de puissance alloué à un utilisateur
$P_{u,n}$	:	puissance affectée à la porteuse n de l'utilisateur u
$\gamma_{u,n}$	:	rapport signal à bruit de l'utilisateur u sur la porteuse n
$g_{u,n}$	:	gain de canal (complexe) sur la sous porteuse n vu par l'utilisateur u
$\phi_{u,n}$	:	CgNR de la sous porteuse n de l'utilisateur u ($ g_{u,n} ^2 / N_0 \Delta f$)
$b_{u,n}$	:	nombre de bits par symbole sur la porteuse n de l'utilisateur u (paramètre de <i>bit loading</i> , $b_{u,n}$ est un entier)
$r_{u,n}$	:	débit sur la porteuse n de l'utilisateur u ($r_{u,n} \in \mathbb{R}$)
r_u	:	débit de l'utilisateur u sur $\mathcal{Q}_u$
r_u°	:	contrainte sur le débit de l'utilisateur u = débit minimal de u
R_{max}	:	nombre maximal de bits par symbole sur une sous porteuse

Acronymes

ACG	:	<i>Amplitude Craving Greedy algorithm</i>
BABS	:	<i>Bandwidth Assignment based on SNR</i>
BARE	:	<i>Bandwidth Allocation on rate estimation</i>
bDA	:	<i>basic Dynamic Assignment</i>
BER	:	<i>Bit Error Rate</i>
CgNR	:	<i>Channel gain to Noise Ratio</i>
MA	:	<i>Margin Adaptive optimization</i>
MCS	:	<i>Modulation and Coding Scheme</i>
OFDM	:	<i>Orthogonal Frequency Division Multiplexing</i>
OFDMA	:	<i>Orthogonal Frequency Division Multiple Access</i>
QAM	:	<i>Quadrature Amplitude Modulation</i>
RA	:	<i>Rate Adaptive optimization</i>
RPO	:	<i>Rate Profit Optimization</i>
SER	:	<i>Symbol Error Rate</i>
SNR	:	<i>Signal Noise To Ratio</i>

Résumé

Dans ce document, nous nous intéressons, dans un contexte OFDMA¹, à l'allocation de ressources (sous porteuses et puissance) dans le sens descendant d'une cellule radio isolée ; nous ne considérons donc pas d'interférences co-canal.

En OFDMA, l'ensemble des sous porteuses d'un symbole OFDM² est divisé en sous ensembles, ces derniers peuvent être affectés à différents utilisateurs. L'OFDMA apporte une nouvelle dimension lors de l'allocation des ressources : il faut affecter les différentes fréquences aux utilisateurs pendant un même *time slot*. De plus, il faut déterminer un MCS³ pour chaque fréquence et cela dépend de l'utilisateur auquel elle est affectée. Les possibilités d'allocation sont multiples. C'est pourquoi, plusieurs travaux ont été réalisés sur ce sujet depuis 1999. Différents objectifs peuvent être poursuivis lors de l'allocation de ressources : certains cherchent à minimiser la puissance transmise dans la cellule, d'autres à maximiser le débit total de la cellule.

Nous proposons une synthèse sur les travaux existants : nous exprimons les différentes formulations des problèmes d'optimisation puis les techniques adoptées pour les résoudre.

D'autre part, nous apportons une contribution dans la résolution du problème suivant : maximiser le débit global de la cellule sous des contraintes de débit individuelles. Cette contribution a par ailleurs fait l'objet d'une publication dans la conférence VTC⁴ ([Lengoumbi&_06]). L'algorithme proposé concerne l'attribution des sous porteuses. La puissance est supposée uniformément répartie sur l'ensemble des sous porteuses. Nous comparons l'algorithme proposé à des algorithmes d'attribution de sous porteuses décrits dans notre synthèse.

¹ *Orthogonal Frequency Division Multiple Access*

² *Orthogonal Frequency Division Multiplexing*

³ *Modulation and Coding Scheme*

⁴ *Vehicular Technology Conference*

1. Introduction

1.1. Contexte de l'OFDMA

L'OFDM (*Orthogonal Frequency Division Multiplexing*) est une modulation à porteuses multiples dont la principale caractéristique est l'orthogonalité des canaux supportés par chaque porteuse (appelée aussi sous porteuse). Les modulations à porteuses multiples ont été introduites pour combattre l'interférence entre symboles (et donc simplifier la tâche des égalisateurs) tout en maintenant un débit élevé. Le flux numérique de débit de rythme R (bit/s) est multiplexé en N flux parallèles de rythme R/N . Les N porteuses sont orthogonales car espacées de $1/T_s$ où T_s est la durée d'un symbole OFDM ([ShinsukePrasad_03]). Les effets dispersifs du canal sont combattus grâce à l'utilisation d'un intervalle de garde (orthogonalité temporelle). L'orthogonalité fréquentielle est préservée par le contenu de l'intervalle de garde : un préfixe, dit cyclique, permet de diagonaliser la matrice de gain du canal dans l'espace de Fourier ([Ciblat_04]). La fin de la partie utile du symbole OFDM est recopiée dans ce préfixe cyclique. Finalement, la transmission sur chaque sous porteuse subit simplement un facteur multiplicatif (correspondant à la réponse fréquentielle du canal) et un bruit additif. En OFDM, la répartition de la puissance peut être optimisée (*power loading*, [Wyglinski_04]) ainsi que la taille de la constellation sur chaque sous porteuse (*bit loading*, [Czylwik_96]). Parmi les modulations à porteuses multiples, l'OFDM est largement utilisée ; on peut citer le DAB (Digital Audio Broadcasting), DVB-T (Digital Broadcasting Video Terrestrial), IEEE 802.11a et 802.16.

Dans un contexte multi-utilisateur, l'utilisation de l'OFDM est couplée à celle des techniques d'accès multiples. En OFDM-TDMA un utilisateur reçoit toutes les fréquences du symbole OFDM durant un slot et les mauvaises fréquences ne sont pas utilisées ; elles sont alors perdues pendant la durée du slot. L'OFDMA, *Orthogonal Frequency Division Multiple Access*, est introduite pour assurer une meilleure gestion du spectre. La bande est divisée en sous canaux qui sont attribués à différents utilisateurs. Cela permet de profiter d'une diversité fréquentielle grâce à laquelle une mauvaise fréquence pour un utilisateur est susceptible d'être bonne pour un autre. Concernant les techniques de construction de sous canaux, on distingue les techniques de saut de fréquence aléatoire ([RadioCom_04], [Lawrey_99]), de répartition distribuée sur la bande ([IEEE802.16]) et d'ajustement spécifique à chaque utilisateur (de nombreuses références seront citées au cours de ce document). Concernant cette dernière méthode de construction de sous canaux, nous étudierons des algorithmes d'allocation de ressources dans le sens descendant d'une cellule sans considérer d'interférences co-canal. Les problèmes formulés dans ce contexte sont exprimés dans la section suivante.

1.2. Problèmes d'optimisation avec contrainte en OFDMA

L'affectation des ressources en OFDMA est réalisée grâce à la résolution de problèmes d'optimisation avec contraintes. Ci-après nous énumérons les différents types de problèmes rencontrés.

1.2.1 Minimiser la puissance émise

Un objectif est ici de garantir un débit r_u° (et éventuellement une qualité de service exprimée en BER) à chaque utilisateur u , $1 \leq u \leq U$, tout en minimisant la puissance totale émise $P_T(\underline{r}) = \sum_u P_u(r_u)$. On écrit :

$$\begin{aligned} & \min_{\underline{r}} P_T(\underline{r}) \\ & \text{avec } \underline{r} \geq \underline{r}^\circ \\ & \text{où } \underline{r} = (r_1, \dots, r_u, \dots, r_U) \text{ et } \underline{r}^\circ = (r_1^\circ, \dots, r_u^\circ, \dots, r_U^\circ) \end{aligned}$$

Ce problème est connu sous le nom de *Margin Adaptive optimization* (MA). L'expression de la puissance $P_T(\underline{r})$ dans la formulation ci-dessus est décrite dans l'annexe A.

1.2.2 Maximiser le débit

Un autre objectif peut être maximiser le débit total des usagers avec comme contraintes un budget de puissance fixé et éventuellement des débits minimaux r_u° pour chaque utilisateur u . L'expression du problème est alors :

$$\begin{aligned} & \max \sum_u r_u \\ & \text{avec } P_T < P_{T,max}, \quad (1) \\ & \text{et } \underline{r} \geq \underline{r}^\circ \quad (2) \end{aligned}$$

Ce problème est connu sous le nom de *Rate Adaptive optimization* (RA). La contrainte (2) peut ne pas être considérée ([JanLee_03], [KimHan&_04]), ou exprimée différemment afin d'imposer des conditions supplémentaires sur les débits finaux des utilisateurs ([Wong&_04]). L'impact de la contrainte (2) est étudié dans [ChangKuo_04].

Le calcul du débit d'un utilisateur s'obtient par une fonction $r = h(\gamma)$ qui lie le débit r avec le SNR γ . Le tableau 1 résume quelques possibilités pour la fonction $h(\cdot)$. Dans ce tableau, Γ désigne le *SNR gap* (il correspond à une marge de puissance, cf. annexe A). La fonction $h(\cdot)$ communément adoptée est la formule de Shannon ; elle est souvent remplacée par d'autres qui tentent de prendre en compte les MCS (*Modulation and Coding Scheme* ou schémas de modulation et de codage) utilisés par le système considéré (cf. Fig. 1).

Remarques : a) Le problème de maximisation du débit peut également s'exprimer avec une contrainte sur le taux d'erreur bit moyen des sous porteuses (comme [Wygliniski_04], dans le cadre d'un unique utilisateur en OFDM).

b) Avec les contraintes (1) et (2), certains auteurs cherchent à minimiser la quantité de ressources fréquentielles utilisée ([Koutsopoulos_02], [Acena_05]).

Tableau 1: Fonctions "puissance-débit"

$h(\gamma) = \log_2(1 + \gamma/\Gamma)$	variante : $\min(R_{max} \log_2(1 + \gamma/\Gamma))$
$h(\gamma) = \min(R_{max}, f^{-1}(\gamma))$ où $\gamma = f(r)$; on a $f(r) = 0.6 r^3$ dans [Kivanc&_03]	

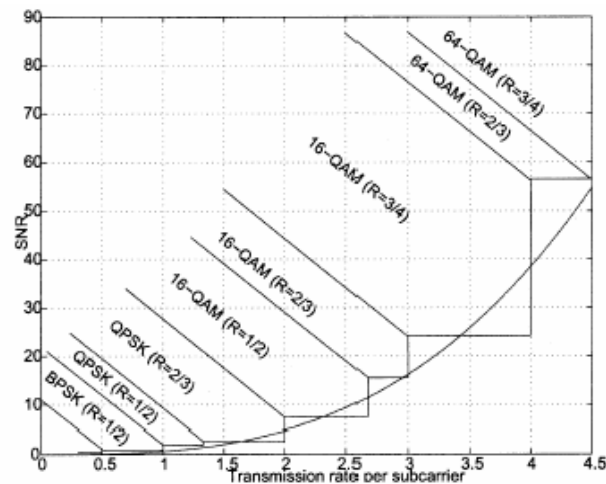


Fig. 1. $f(r)$ for $P_e = 1 \times 10^{-6}$ and continuous approximation $f(r) = 0.60r^3$.

Figure 1: Fonction "débit-puissance" considérée dans [Kivanc&_03]

1.2.3 Introduire l'équité entre les utilisateurs

Il existe plusieurs sortes d'équité, on peut citer l'équité en bande et l'équité en débit.

L'équité en bande assure aux utilisateurs une quantité égale de ressources fréquentielles. Le défaut de ce type d'équité, réside dans la disparité des conditions radio des usagers.

L'équité en débit garanti aux usagers des débits similaires indépendamment de leurs conditions radio. En MA, l'équité en débit se traduit par $r_u^o = r^o$ pour tout u dans la contrainte sur le vecteur de débit (cf. 1.2.1). Pour assurer une équité en débit en RA, on cherche à maximiser le débit minimal que l'on peut garantir à tout utilisateur. Cela revient à résoudre le problème suivant ([RheeCioffi_00]):

$$\max_A (\min_u (r_{u,A}))$$

avec $P_T < P_{T,max}$

où A est une instance d'affectation en fréquence et puissance.

2. Algorithmes d'allocation des ressources

Un problème majeur des algorithmes est relatif à leur temps d'exécution c'est-à-dire à leur complexité. Sans simplifications préalables, les problèmes d'affectation conjointe (optimale) de sous porteuses et de puissance sont réputés difficiles et qualifiés NP complets ([Cheng&_04], [Anas_04]).

Dans les algorithmes d'allocation de ressources, on distingue les techniques de résolution « directe » et les techniques de résolution « par étapes ».

Par résolution « directe », on entend l'optimisation conjointe des différents paramètres (sous porteuses, modulation et/ou puissance) ; tandis qu'une résolution « par étapes » considère séparément l'attribution des sous porteuses et le choix des MCS (*Modulation and Coding Scheme*). Un exemple de résolution directe est proposé par [Inhyoung&_01]. Les auteurs proposent une forme linéaire du problème d'optimisation à variables entières. A l'origine la fonction de coût et les contraintes sont non linéaires. Le problème ainsi reformulé est linéaire (fonction de coût et contraintes). Il peut être résolu par des algorithmes de programmation entière (ou *integer programming*). Cette technique fournit la solution optimale ([Ergen&_03]). Cependant, la complexité croît de façon exponentielle avec le nombre de contraintes et de variables.

Le défi porte sur l'efficacité des algorithmes. Un but est de pouvoir attribuer les ressources en temps réel en faisant face aux variations du canal. C'est pourquoi, généralement, les auteurs proposent une résolution « par étapes » ; l'optimisation d'étapes distinctes simplifie le problème.

Si la décomposition de l'allocation des ressources en plusieurs étapes peut parfois aboutir, dans des cas « simples »¹, à une solution optimale, généralement la solution obtenue est sous optimale. L'objectif consiste à se rapprocher des performances de la solution optimale tout en améliorant la rapidité d'exécution.

2.1. Décomposition par « étapes » dans les problèmes MA

Lors de la résolution du problème MA, on rencontre deux méthodes (respectivement Fig.2 et Fig.3).

2.1.1 Première méthode : Décomposition 1

Dans la première méthode (cf. Fig.2), on rencontre deux étapes qui sont l'attribution des sous porteuses et le choix du MCS sur chaque sous porteuse (en cas de modulation adaptative). Cette méthode est appliquée dans [WongMurch_99] et [Inhyoung&_01].

L'attribution des sous porteuses (étape 1, Fig.2) peut se décomposer en 2 tâches. La tâche 1 détermine le nombre N_u de sous porteuses pour chaque utilisateur u . Les techniques de résolution de la tâche 1 sont détaillées dans la section 3. Une fois la tâche 1 effectuée, la tâche 2 peut être effectuée de façon optimale² par l'algorithme Hongrois (cf. 4.1) ou par des heuristiques décrites en 4.2.

¹ Il s'agit du problème MA avec modulation fixe ([WongTsui_99], [PietrzickJan_02]) ou du problème RA sans contraintes de débits ([JanLee_03], [KimHan&_04]).

² Pour une répartition $\{N_u\}_{1 \leq u \leq U}$ donnée.

Soit Ω_u , l'ensemble des sous porteuses affectées à l'utilisateur u à la fin de la tâche 2. Le choix des MCS sur les sous porteuses est effectuée (étape 2, Fig.2) de façon à minimiser la puissance de transmission requise. L'algorithme optimal réalisant cette étape est connue sous le nom de *bit loading*.

Description de l'algorithme de *bit loading* (étape 2, décomposition 1 en MA)

Chaque utilisateur u est traité séparément. Sur l'ensemble Ω_u , le nombre¹ de bits $b_{u,n}$ transportés sur la sous porteuse n est incrémenté sur la sous porteuse qui génère la plus faible augmentation de puissance transmise. Le procédé s'arrête lorsque la somme des bits $b_{u,n}$ des sous porteuses n de Ω_u atteint le débit cible r_u^o .

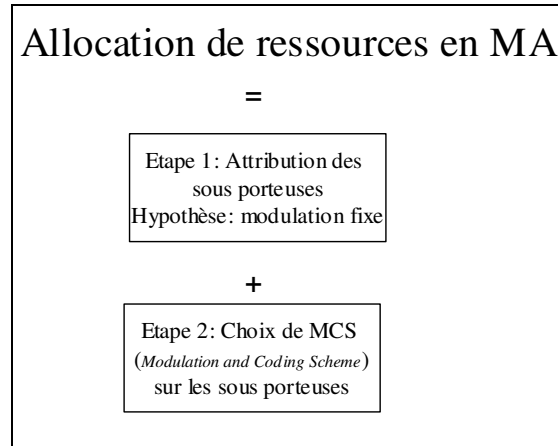


Figure 2: Décomposition 1 en MA ([WongMurch_99], [Inyoung&_01])

2.1.2 Deuxième méthode : Décomposition 2

La seconde méthode en MA (cf. Fig.3), consiste à introduire deux phases dont la première est dite grossière ([WongTsui_99], [PietrykJan_02], [Ergen_03], [Chen&_04]). La première phase réalise une affectation rapide, concernant à la fois les sous porteuses et les choix de MCS. La seconde phase est une phase d'amélioration dont l'objectif est de réduire la puissance transmise. L'intérêt de cette deuxième méthode est de pouvoir s'adapter à la vitesse de variations du canal. Lorsque le canal varie vite², seule l'allocation grossière est faite ; si le canal varie plus lentement, la phase d'amélioration peut affiner l'allocation.

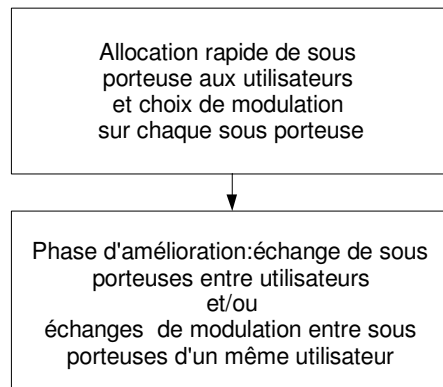


Figure 3: Décomposition 2 en MA ([WongTsui_99], [PietrykJan_02], [Ergen_03], [Chen&_04])

¹ Dans ce document, nous utilisons $b_{u,n}$ pour des entiers et $r_{u,n}$ pour des réels. Dans la littérature l'algorithme du *bit loading* est utilisé avec des nombres de bits entiers par symboles QAM (*Quadrature Amplitude Modulation*).

² C'est-à-dire pour de faibles valeurs du temps de cohérence : T_c .

Les techniques utilisées pendant la phase d'amélioration sont l'échange de sous porteuses entre deux utilisateurs ou le changement de MCS sur les sous porteuses d'un même utilisateur. Ces procédés sont détaillés en 4.2.2.

2.2. Décomposition par « étapes » dans les problèmes RA

Lors de la résolution du problème RA, il est plus simple de séparer l'attribution de sous porteuses et l'allocation de la puissance (cf. Fig.4). Ces deux étapes sont distinguées dans les références suivantes [Kivanc&_00], [Kivanc&_03], [YinLiu_00], [Wong&_04] et [Anas_04].

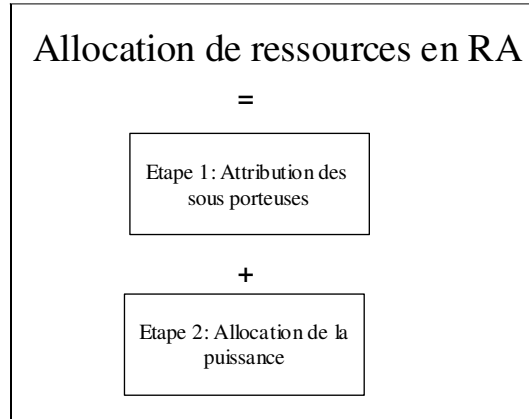


Figure 4: Décomposition en RA ([Kivanc&_00], [Kivanc&_03], [YinLiu_00], [Wong&_04], [Anas_04])

L'étape 1 peut se décomposer en deux tâches (cf. Fig.6): calcul du nombre de sous porteuses N_u pour chaque utilisateur u et affectation spécifique de sous porteuses. Ces tâches sont respectivement traitées dans les sections 3 et 4.

La solution optimale de l'étape 2 (allocation de puissance) est le *waterfilling* lorsque le bruit est gaussien ([CoverThomas_91]). Le *waterfilling* n'est réalisé que lorsque l'attribution des sous porteuses aux utilisateurs (réalisée par l'étape 1) est connue. De même, en MA, l'algorithme du *bit loading* n'est possible que lorsque les ensembles $\{\Omega_u\}_{1 \leq u \leq U}$ sont formés.

Principe du *waterfilling*

Le *waterfilling* consiste à répartir l'énergie en fonction d'un « niveau d'eau » et de l'inverse du gain du canal (cf. Fig. 5). Plus le CgNR (*Channel gain to Noise Ratio*, $\varphi_n = |g_n|^2 / N_0 \Delta f$) d'une sous porteuse est élevé, plus elle recevra de puissance. La puissance est d'autant plus élevée que l'inverse du CgNR est éloigné du niveau d'eau, cela s'exprime par la relation $p_n(\nu) = \nu - \varphi_n^{-1}$ où ν est le niveau d'eau et φ_n le CgNR du canal n (cf. annexe C). Dans le cadre multi-utilisateur, le *waterfilling* est appliqué pour chaque utilisateur u sur l'ensemble Ω_u .

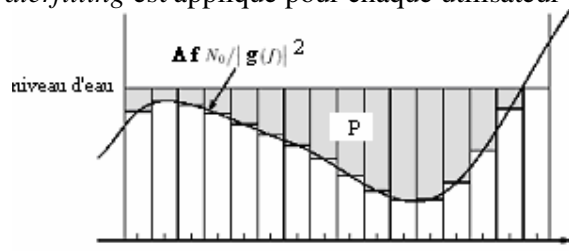


Figure 5: Principe du *waterfilling*

¹ Rappelons que Ω_u est l'ensemble des sous porteuses qui sont affectées à l'utilisateur u .

2.3. BARE et RPO, deux contributions au problème RA

Notre contribution s'applique au problème RA :

Maximiser $\sum_u r_u$

avec $P_T < P_{T,Max}$, (1)

et pour tout u , $r_u \geq r^\circ$ (2a)

ou pour tout u , $r_u \geq r_u^\circ$ (2b)

On considère deux versions (2a) et (2b) de la contrainte (2) sur le vecteur de débit. Dans la version (2a) de la contrainte sur le vecteur de débit, le débit minimal à respecter r° est commun à tous les usagers ; c'est un type d'équité qui peut être qualifié **d'équité en débit minimal**. Dans la version (2b) de la contrainte sur le vecteur de débit, chaque usager u possède une contrainte de débit minimal r_u° qui lui est propre.

Afin de résoudre ce problème (quelque soit la version de la contrainte 2), nous adoptons la décomposition classique : attribution de sous porteuses puis allocation de puissance (cf. Fig.4). Concernant la puissance, nous choisissons une allocation uniforme: chaque sous porteuse reçoit la puissance $P_{T,max}/N$. Ce choix est également effectué dans [RheeCioffi_00], [JanLee_03], [Gross&_03], [Anas_04] et [HuiZhou_06]. Les auteurs de [JanLee_03] montrent que cette stratégie apporte peu de dégradations comparé au *waterfilling*.

Pour traiter l'attribution de sous porteuses, on considère les deux tâches ci-après (cf. Fig.6) :

- calcul du nombre de sous porteuses N_u pour chaque utilisateur u
- affectation spécifique des sous porteuses aux utilisateurs (i.e. construction des ensembles Ω_u où $|\Omega_u| = N_u$).

Elles constituent respectivement les tâches 1 et 2. Nous appelons la tâche 1, allocation de largeur de bande car on y détermine le nombre de sous porteuses de chaque utilisateur. Pour cette tâche 1, nous proposons l'algorithme BARE (*Bandwidth Assignment Based on Rate Estimation*) dans la section 3. Concernant la tâche 2, nous proposons le RPO (*Rate Profit Optimization*) dans la section 4. Dans les sections 3 et 4, nous effectuons au préalable une synthèse sur les algorithmes existants. La partie 5 présente les résultats de simulation sur le BARE et le RPO, la plupart de ces résultats sont présentés dans [Lengoumbi&_06].

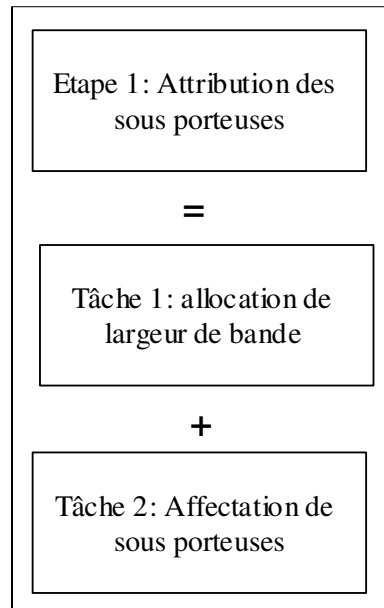


Figure 6: Phase d'allocation de sous porteuses ([Kivanc&_00], [Kivanc&_03], [YinLiu_00], [Gross&_03])

3. Tâche 1 : Allocation de largeur de bande

En présence d'une contrainte sur le débit minimal (qu'il s'agisse d'un problème MA ou RA), la phase d'affectation de sous porteuses est souvent précédée par le calcul du nombre N_u de sous porteuses à affecter à chaque utilisateur u . En modulation fixe, il y a une relation directe entre le débit minimal et le nombre de sous porteuses nécessaires ([WongTsui_99], [PietrykJan_02]). En modulation adaptative, le lien est moins direct et des approches plus ou moins élaborées ont été proposées par plusieurs articles. En 3.1 nous étudions ces approches et en 3.2 nous exposons notre proposition pour la tâche 1 en RA.

3.1. La tâche 1 dans la littérature

Les approches pour résoudre la tâche 1 peuvent être classées selon l'expression de la contrainte sur le vecteur de débit.

3.1.1 Contrainte $\underline{r} \geq \underline{r}^o$

L'algorithme que nous présentons est utilisé aussi bien en MA ([Kivanc&_00], [Kivanc&_03]¹) qu'en RA ([YinLiu_00]), il y diffère dans les conditions d'initialisation et d'arrêt. Soit le vecteur de débits minimaux $\underline{r}^o = (r^o_1, \dots, r^o_u, \dots, r^o_U)$. Une étape préliminaire consiste à calculer le vecteur des CgNR moyens $\underline{\varphi} = (\varphi_1, \dots, \varphi_u, \dots, \varphi_U)$. Dans l'algorithme, on considère que toutes les sous porteuses ont le même CgNR pour un utilisateur donné. Il

s'agit alors du CgNR moyen $\bar{\varphi}_u$ vu par l'utilisateur u : $\bar{\varphi}_u = (1/N) \sum_{n=1}^N \varphi_{u,n}$.

Soit $i(u)$ le nombre de sous porteuses attribué à l'utilisateur u à une étape donnée. La boucle décrite ci-dessous porte sur le vecteur $\underline{i}$. Les actions ci-après sont effectuées de façon itérative :

- pour chaque u , connaissant $\bar{\varphi}_u$, on calcule $P(i(u))$, la puissance nécessaire pour transmettre r^o_u bits avec $i(u)$ sous porteuses,
- de même, on calcule la puissance $P(i(u)+1)$ nécessaire pour transmettre r^o_u bits si l'utilisateur avait $i(u)+1$ sous porteuses,
- on incrémente $i(u)$ pour l'utilisateur qui vérifie le plus grand $\Delta(u) = P(i(u)) - P(i(u)+1)$.

A chaque étape, l'utilisateur u qui reçoit une sous porteuse supplémentaire est celui qui réalise la plus grande économie de puissance.

Calcul de la puissance² $P(i(u))$:

On considère la fonction f qui relie le SNR γ au débit r : $\gamma = f(r)$ ³ (la fonction est différente selon le BER exigé). Alors la l'expression de la puissance nécessaire à u est : $P(i(u)) = (i(u) / \bar{\varphi}_u) \times f(r^o_u / i(u))$. [Kivanc&_03] propose de prendre $f(r) = 0.6r^3$ (BER = 10^{-6}) ; f est une fonction croissante donc $f(r^o_u / i(u))$ décroît quand $i(u)$ augmente ($\Delta(u) \geq 0$). Constatons qu'entre deux usagers u_1 et u_2 , à r^o_u et $i(u)$ identiques, celui qui recevra la sous porteuse supplémentaire vérifie $\varphi_{u1} < \varphi_{u2}$.

¹ L'algorithme MA y est nommé BABS (*Bandwidth Assignment Based on SNR*).

² Puissance nécessaire à u pour réaliser un débit r^o_u avec $i(u)$ sous porteuses.

³ $f \sim h^{-1}$ où h est définie est 1.2.2.

3.1.1.1. Initialisations

On trouve deux façons d'initialiser $i(u)$. Dans [Kivanc&_00] et [Kivanc&_03] (BABS *Bandwidth Assignment Based on SNR*), à la première itération, $i(u) = \lceil r_u^\circ / R_{max} \rceil$ pour chaque utilisateur u ; rappelons que R_{max} est le nombre de bits maximal par symboles QAM. Après cette initialisation, si le nombre total de sous porteuses dépasse N , les utilisateurs de plus faible débit minimal sont éliminés¹. Dans [Nakad_03], on propose le contraire afin de servir un plus grand nombre d'utilisateurs².

Dans [YinLiu_00] (problème RA), à la première itération $i(u) = 1$ pour chaque utilisateur u ; les auteurs ne prennent pas en compte de R_{max} .

3.1.1.2. Condition d'arrêt

Dans le BABS ([Kivanc&_00], [Kivanc&_03]), la condition d'arrêt est l'utilisation de toutes les porteuses i.e. $\sum_u i(u) = N$. Les auteurs montrent que la répartition $\{N_u\}_{1 \leq u \leq U}$ obtenue par le BABS est optimale³ dans le cas où chaque utilisateur voit toutes les sous porteuses avec le même CgNR.

La condition d'arrêt de [YinLiu_00] fait intervenir $P_{T,max}$ car on y résout un problème RA. Cette condition d'arrêt⁴ est : $P_T / N_a \leq P_{T,max} / N$. Rappelons que P_T est la somme des puissances allouées à un instant donné, N_a est le nombre total de sous porteuses attribuées aux usagers à une étape donnée, $P_{T,max}$ est la contrainte de puissance totale et N est le nombre de sous porteuses disponibles.

La valeur définitive du nombre de sous porteuses de l'utilisateur u est notée N_u (à distinguer avec les valeurs temporaires $i(u)$ au cours de l'algorithme).

3.1.1.3. Autres propositions

Il existe une version de l'algorithme décrit en 3.1 pour la voie montante ([Pfletschinger&_02]). Chaque utilisateur a désormais une contrainte de puissance qui lui est propre. L'initialisation est $i(u) = \lceil r_u^\circ / R_{max} \rceil$. Si le nombre total de sous porteuses n'est pas atteint, on incrémente $i(u)$ pour les usagers qui dépassent leur contrainte de puissance jusqu'à ce qu'ils ne la dépassent plus. Puis tant que le nombre total de sous porteuses est dépassé, l'utilisateur qui minimise l'augmentation de puissance se voit retirer une sous porteuse⁵.

Dans [KimKwak_04], les nombres définitifs de sous porteuses $\{N_u\}_{1 \leq u \leq U}$ ne sont pas connus à l'avance. Les auteurs posent $i(u) = \lceil r_u^\circ / R_{max} \rceil$ pour pouvoir effectuer l'allocation initiale de sous porteuses, les nombres définitifs ($N_u = |\mathcal{Q}_u|$) sont connus à la fin de l'allocation dite résiduelle (cf. 4.2.2).

¹ Avec l'initialisation du BABS, la somme des composantes du vecteur i peut dépasser N . Pour y remédier les auteurs proposent un patch final : une boucle « while » de condition $\sum_u i(u) > N$ qui supprime les fréquences ($i(u) := 0$) à celui qui a le plus faible $i(u)$. Bien sûr, on doit travailler avec $\min(i(u)^+)$ ($i(u)^+$ est l'ensemble des $i(u)$ strictement positifs) et non $\min(i(u))$, sinon la boucle est infinie.

² Elimination d'un utilisateur à débit minimal élevé au lieu de plusieurs utilisateurs de faible débit minimal.

³ C'est-à-dire que la répartition $\{N_u\}_{1 \leq u \leq U}$ minimise la puissance transmise.

⁴ On pourrait tout à fait envisager une condition d'arrêt plus simple directement inspirée de la contrainte en RA : $P_T \leq P_{T,max}$. Cependant, avec la condition : $P_T < P_{T,max} (N_a / N)$, l'algorithme s'arrête plus tard qu'avec la condition $P_T < P_{T,max}$; en effet, $N_a / N < 1$ et la puissance P_T décroît au cours de l'algorithme (les utilisateurs ont de plus en plus de sous porteuses). Comme l'algorithme s'arrête plus tard, le nombre total de sous porteuses utilisées est plus important qu'avec la simple condition d'arrêt $P_T < P_{T,max}$. Cela est cohérent avec l'idée de maximiser le débit sous une contrainte de puissance donnée (ce qui est le but final de l'algorithme dans [YinLiu_00]).

⁵ Il se peut que l'utilisateur qui minimise l'augmentation de puissance dépasse sa contrainte de puissance si on lui retire une sous porteuse, or l'algorithme lui retire une sous porteuse sans vérifier cette condition.

3.1.2 Rapports de proportionnalité dans la contrainte sur le débit

Le problème RA à résoudre dans [Wong&_04] et [HuiZhou_06] est exprimé avec une variante de la contrainte (2) définie en 1.2.2. En effet, à la fin de l'affectation des ressources, le vecteur $\underline{r}$ doit vérifier pour tout couple d'utilisateurs (u, v) $r_u/r_v = \beta_u/\beta_v$ où $\underline{\beta}$ est donné en entrée¹. Le nombre de sous porteuses est déterminé comme suit : $i(u) = \lfloor \beta_u \cdot N \rfloor$. La somme des porteuses allouées à ce stade n'est pas nécessairement égale à N . Les sous porteuses restantes après une première phase d'allocation sont allouées de façon opportuniste (cf. 4.3.2).

En MA, [Chen&_04] utilise aussi les rapports entre les débits finaux à réaliser pour initialiser $i(u)$. Les nombres définitifs N_u peuvent être différents car une phase d'amélioration (échange et redistribution des sous porteuses) a lieu après l'affectation des sous porteuses.

3.2. Une contribution pour la tâche 1 en RA : le BARE

3.2.1 Introduction

Nous proposons un algorithme pour déterminer le nombre de sous porteuses (ou largeur de bande) par utilisateurs en RA. L'expression de la contrainte sur les débits est $\underline{r} \geq \underline{r}^o$ où $\underline{r}^o$ est un vecteur quelconque.

La motivation de cet algorithme est la suivante.

L'algorithme BABS (cf. 3.1) cherche à minimiser des puissances, il alloue une sous porteuse supplémentaire à l'utilisateur qui va utiliser le moins de puissance. Cette approche est plus adaptée au cas MA qu'au cas RA. Nous nous intéressons au cas RA, et adoptons l'hypothèse d'allocation uniforme de puissance sur les sous porteuses. Ainsi nous proposons un algorithme BARE (*Bandwidth Allocation on Rate Estimation*) qui alloue une sous porteuse supplémentaire sur un critère de débit et non de puissance.

3.2.2 Description

Soit $\bar{\varphi} = (\bar{\varphi}_1, \dots, \bar{\varphi}_u, \dots, \bar{\varphi}_U)$. le vecteur de CgNR moyens. D'après la même simplification que le BABS, on considère qu'un utilisateur u voit toutes les sous porteuses avec le même CgNR moyen $\bar{\varphi}_u$. Lorsqu'un utilisateur u reçoit une largeur de bande de M sous porteuses, son débit est estimé à $r_{est}(u, M) = M \times h(p, \bar{\varphi}_u)$ car $h(p, \bar{\varphi}_u)$ est le débit estimé d'une sous porteuse de u (rappelons que $p = P_{T,max}/N$).

Le vecteur i représente les nombres de sous porteuses (ou largeur de bande) des utilisateurs au cours de l'algorithme. Dans notre algorithme, nous initialisons le nombre de sous porteuses à $i(u) = \lfloor N/U \rfloor$.

L'algorithme cherche à minimiser la différence entre le débit estimé et le débit minimal de chaque utilisateur. Pour cela, l'algorithme modifie la largeur de bande $i(u)$ qui est attribuée à un utilisateur u .

La différence entre le débit estimé et le débit minimal recherché est appelée *gap*. Pour un utilisateur u , cette différence dépend de la largeur de bande $i(u)$ reçue: $gap(u, i(u)) = r_{est}(u, i(u)) - r_u^o$.

¹ On peut parler de *fair proportional constraints* ou d'équité pondéré.

Si les N sous porteuses ne sont pas toutes utilisées à la fin de l'initialisation¹ du vecteur $\underline{i}$, on lance la phase *post_initialisation*. Quand les N sous porteuses sont toutes utilisées, on lance la phase *principale*.

3.2.2.1. Phase post_initialisation

Tant que les N sous porteuses ne sont pas utilisées², l'utilisateur le plus éloigné de son objectif (c'est-à-dire celui qui a le *gap* le plus faible) reçoit une sous porteuse supplémentaire.

```
while  $\sum_{u=1}^U i(u) < N$ ,
     $i(u^*) = i(u^*) + 1$  where  $u^* = \arg \min_u \text{gap}(u, i(u))$ .
end while
```

3.2.2.2. Phase principale de l'algorithme BARE

S'il existe un utilisateur u^* qui ne satisfait pas son objectif et un utilisateur u' qui satisfait son objectif malgré le retrait d'une sous porteuse, on retire une sous porteuse à u' et on l'ajoute à u^* et ainsi de suite.

```
while  $\min_u (\text{gap}(u, i(u)) < 0$  and  $\exists u'$  such as  $\text{gap}(u', i(u') - 1) > 0$ ,
    -  $i(u^*) = i(u^*) + 1$  where  $u^* = \arg \min_u \text{gap}_u(u, i(u))$ 
    -  $i(u^\circ) = i(u^\circ) - 1$  where  $u^\circ = \arg \max_u \text{gap}_u(u, i(u) - 1)$ 
end while
```

L'algorithme se termine avec succès lorsque $\min_u (\text{gap}(u, i(u))) \geq 0$. Dans le cas contraire (c'est-à-dire $\min_u (\text{gap}(u, i(u))) < 0$), on dit que l'algorithme BARE n'abouti pas. Les débits minimaux des utilisateurs qui vérifient $\text{gap}(u, i(u)) < 0$, ont peu de chances d'être satisfaits après l'affectation des sous porteuses (tâche 2). Cela provoque un *outage*³ élevé.

Quand l'algorithme BARE n'abouti pas, deux solutions existent pour éviter un *outage* élevé à la fin de la tâche 2: augmenter le budget de puissance ou baisser les exigences sur le vecteur de débit minimal. Ci-après, on étudie les possibilités d'actions sur le vecteur de débit minimal pour obtenir une contrainte réalisable, c'est-à-dire une contrainte de débit pour laquelle l'algorithme BARE se termine avec succès.

3.2.3 Evolutions de la contrainte sur le vecteur de débit

Rappelons le problème que l'on cherche à résoudre :

Maximiser $\sum_u r_u$

avec $P_T < P_{T,Max}$, (1)

et pour tout u , $r_u \geq r^\circ$ (2a)

ou pour tout u , $r_u \geq r_u^\circ$ (2b)

L'algorithme BARE se prête aussi bien à la résolution du problème 1 (contrainte (2a)) qu'à celui du problème 2 (contrainte (2b)). Rappelons que nous adoptons une allocation uniforme de puissance sur les sous porteuses.

¹ Cela correspond au cas où N n'est pas un multiple de U .

² La somme des composantes du vecteur i est différente de N .

³ On appelle *outage* la proportion d'utilisateurs qui ne vérifient pas leur débit minimal à la fin de la tâche 2.

3.2.3.1. Recherche d'une contrainte (2a) réalisable

La contrainte est commune à tous les utilisateurs : le vecteur de débit minimal est $\underline{r}^\circ = r^\circ \underline{1}$. Cela correspond à une équité en débit minimal.

Pour ce problème, l'algorithme BARE est appliqué avec $r_u^\circ = r^\circ$ pour tous les utilisateurs. Lorsque la contrainte r° est très élevée, le problème peut ne pas avoir de solutions¹. Dans ce cas, l'algorithme BARE n'abouti pas (i.e. $\min_u(gap_u(N_u)) < 0$)².

Si ce qui importe c'est de trouver un débit minimal commun (et non la valeur de ce débit minimal commun), on peut baisser la contrainte r° d'un facteur λ . Quand l'algorithme BARE se termine sur un échec ($\min_u(gap_u(N_u)) < 0$), le débit minimal commun est décrémenté d'un facteur λ ($\lambda \in \mathfrak{R}^*$).

```
While ( $\min_u(gap(u, N_u)) < 0$ ) & ( $r^\circ > \lambda$ )
     $r^\circ = r^\circ \cdot \lambda$ 
    BARE
end
```

3.2.3.2. Recherche d'une contrainte (2b) réalisable

Le vecteur $\underline{r}^\circ$ est quelconque. Si les valeurs du vecteur $\underline{r}^\circ$ sont trop grandes, le problème peut ne pas avoir de solutions³. Dans ce cas, l'algorithme BARE n'abouti pas (i.e. $\min_u(gap_u(N_u)) < 0$).

Si ce qui importe c'est de trouver un vecteur de débit minimal (et non les valeurs du vecteur), on peut relancer l'algorithme BARE avec $\underline{r}^\circ = \lambda' \underline{r}^\circ$ avec $\lambda' < 1$. Le facteur λ' permet de respecter la structure⁴ initiale du vecteur $\underline{r}^\circ$ et de réduire les valeurs des débits minimaux à satisfaire.

```
While ( $\min_u(gap(u, N_u)) < 0$ ) & ( $\min_u(\underline{r}^\circ) > r_{min}^\circ$ )5)
     $\underline{r}^\circ = \lambda' \underline{r}^\circ$ 
    BARE
end
```

¹ Aucune affectation de sous porteuses, pendant la tâche 2, n'assure aux usagers un débit supérieur au seuil r° .

² A la fin de l'algorithme BARE, on peut parler de N_u et non plus de $i(u)$ qui sont les valeurs au cours de l'algorithme.

³ Aucune affectation de sous porteuses, pendant la tâche 2, n'assure aux usagers un débit supérieur au seuil r_u° .

⁴ Les rapports entre les débits minimaux des différents utilisateurs sont respectés.

⁵ Cette condition supplémentaire fixe un débit seuil à respecter concernant le débit minimal d'un utilisateur.

4. Tâche 2 : Affectation des sous porteuses

4.1. Algorithme Hongrois

L'affectation de sous porteuses est résolue de façon optimale par l'algorithme Hongrois par rapport à un ensemble $\{N_u\}_{1 \leq u \leq U}$ fixé. Cet algorithme est utilisé dans de nombreux domaines car il résout les problèmes d'affectation avec minimisation d'un coût. Le problème peut s'exprimer sous une forme matricielle où chaque terme $c(u,n)$ contient le coût associé à l'affectation de la sous porteuse n à l'utilisateur u . La matrice de coût doit être carrée. Généralement, on dispose de N sous porteuses et U utilisateurs avec $N > U$. Pour appliquer l'algorithme Hongrois, il faut dupliquer chaque utilisateur en N_u utilisateurs fictifs où N_u est le nombre de sous porteuses qu'un usager va recevoir. Chaque usager fictif a les mêmes coûts que l'usager initial. Les usagers fictifs reçoivent une unique sous porteuse à la fin du processus d'affectation. Pour obtenir la solution du problème initial, il suffit de regrouper les sous porteuses des usagers fictifs issus d'un même usager initial. Le fonctionnement de l'algorithme Hongrois est décrit dans l'annexe B. L'ensemble $\{N_u\}_{1 \leq u \leq U}$ détermine la matrice de coût. Si le calcul de cet ensemble est approximatif, la répartition de sous porteuse résultante sera sous optimale.

Choix de la fonction de coût de l'algorithme Hongrois

L'optimalité de l'algorithme Hongrois par rapport au problème initial dépend des nombres N_u et de l'expression de la fonction de coût. L'algorithme Hongrois donne la solution optimale par rapport à la matrice de coût, mais cette dernière est plus ou moins pertinente au regard du problème initial. Etant donné que le coût total doit être minimisé, les possibilités de fonctions de coût sont les suivantes :

- $c(u,n) = -\varphi_{u,n}$ (1)
- $c(u,n) = -10 \log_{10}(\varphi_{u,n})$ ([YinLiu_00]) (2)

Les fonctions de coût (1) et (2) sont optimales lorsqu'on résout respectivement les problèmes $\sum_{u,n} \varphi_{u,n}$ et $\max_{u,n} \sum_{u,n} 10 \log_{10}(\varphi_{u,n})$. Elles peuvent être utilisées aussi bien en MA qu'en RA¹.

- $c(u,n) = -\log_2(1 + P_{u,n} \varphi_{u,n})$ (3)
- $c(u,n) = P_{u,n}$ ([PietrykJan_02]) (4)

La fonction de coût (3) peut être utilisée en RA quand on cherche à maximiser le débit global ($\sum_{u,n} \log_2(1 + P_{u,n} \varphi_{u,n})$). La fonction de coût (4) peut être utilisée en MA quand on cherche à minimiser la puissance transmise. Un problème se pose lorsqu'on considère les fonctions de coût (3) et (4). Pour exprimer l'une ou l'autre de ces fonctions de coût, il faut connaître l'allocation de puissance. Donc pour effectuer la tâche 2 de façon optimale, il faut connaître l'allocation de puissance. Or nous avons vu en 2.2 que l'allocation de puissance (ou le choix des MCS) n'est possible qu'après de la tâche 2. Le problème « se mord la queue ». Pratiquement, pour exprimer la fonction de coût (3) (respectivement (4)) et réaliser la tâche 2, on fait l'hypothèse d'allocation uniforme de puissance sur les sous porteuses (respectivement MCS fixe²).

En conclusion, si l'ensemble $\{N_u\}_{1 \leq u \leq U}$ est optimal et que la stratégie d'allocation de puissance respecte l'hypothèse utilisée pour exprimer la fonction de coût, l'affectation de sous porteuses proposée par l'algorithme Hongrois est optimale au regard du problème initial.

¹ Maximiser le CgNR $\varphi_{u,n}$ de chaque sous porteuse profite aussi bien à la minimisation de la puissance qu'à la maximisation du débit.

² Si le MCS est fixe sur chaque sous porteuse, la puissance nécessaire pour assurer le BER exigé pour chaque utilisateur est alors connue.

L'algorithme Hongrois est réputé lent pour une résolution temps réel (il est donné $\mathcal{O}(N^3)$ dans [Gross&_03] et [DraHen_04]). Plusieurs heuristiques sont proposées dans la littérature pour améliorer le temps d'exécution de l'affectation de sous porteuses. En 4.2 et 4.3 nous présentons les algorithmes rencontrés dans la littérature tandis qu'en 4.4 nous décrivons notre proposition RPO (*Rate Profit Optimization*) pour l'affectation de sous porteuses.

4.2. Algorithmes heuristiques en MA

On peut distinguer les algorithmes qui procèdent par usager ([WongTsui_99], [PietrykJan_02] [Cheng&_04]) et ceux qui procèdent par sous porteuses ([Ergen_03], [Kivanc&_03], [KimKwak_04]). La première catégorie considère un usager et recherche les sous porteuses à lui affecter tandis que la seconde catégorie considère une sous porteuse et cherche à quel usager l'attribuer.

4.2.1 Traitement par usager

Un processus est fréquemment utilisé : il s'agit du classement des sous porteuses réalisé pour chaque usager. Les sous porteuses d'un utilisateur sont ordonnées par gain (ou CgNR) décroissant ([WongTsui_99], [Cheng&_04]) ou par puissance croissante ([PietrykJan_02]). L'ordre de traitement des utilisateurs est arbitraire (par index croissant pour une simplifier la lecture). Soit une matrice $N \times U$, nommée $\mathbf{A}$, comportant des index de sous porteuses. Chaque colonne u est ordonnée de la sous porteuse présentant la meilleure performance pour l'utilisateur u à la moins bonne. Les nombres N_u (fournis par la tâche 1) sont utilisés lors de l'affectation de sous porteuses. Chaque ligne n est parcourue de la colonne 1 à U : la sous porteuse $\mathbf{A}(n,u)$ est affectée à u à condition que la sous porteuse $\mathbf{A}(n,u)$ ne soit pas déjà allouée et que N_u ne soit pas atteint. On peut remarquer que la gestion des collisions découle du caractère arbitraire de l'ordre de traitement des utilisateurs. Le premier utilisateur est privilégié par rapport au dernier : une meilleure sous porteuse commune sera allouée au premier utilisateur plutôt qu'au dernier (à 1 plutôt qu'à U).

Une fois la matrice parcourue ligne par ligne, une phase d'amélioration a pour objectif de minimiser la puissance émise en échangeant les sous porteuses entre les utilisateurs et, si le MCS n'est pas fixe, en modifiant les MCS sur les sous porteuses. A notre connaissance, l'initiateur du procédé d'échange de sous porteuse est [WongTsui_99], cette idée est ensuite reprise dans plusieurs articles ([PietrykJan_02], [Ergen_03], [Cheng&_04]).

Le procédé d'amélioration de la puissance transmise dans [WongTsui_99] est le suivant. Pour chaque couple (u,v) , on calcule $p_{u,v}$ le facteur de réduction de puissance : $p_{u,v} = \delta p_{u,v} + \delta p_{u,v}$. La quantité $\delta p_{u,v}$ est la plus grande réduction de puissance calculée¹ en faisant successivement l'hypothèse que chacune des porteuses de l'utilisateur u est réallouée à v . On nomme $n_{u,v}$ l'index de la sous porteuse de u sur laquelle le maximum $\delta p_{u,v}$ est réalisé. Si $\max(p_{u,v}) > 0$ (maximum pris sur l'ensemble des couples (u,v)), on réalise l'échange : l'utilisateur v reçoit la sous porteuse $n_{u,v}$ et l'utilisateur u reçoit la sous porteuse $n_{v,u}$. Pratiquement, cette tâche peut prendre du temps. On la poursuit tant que la durée de l'allocation ne dépasse pas le temps de cohérence et que $\max(p_{u,v}) > 0$. Quand $\max(p_{u,v}) < 0$, alors aucune amélioration n'est possible. [Cheng&_04] introduit, en plus des échanges, la possibilité de réaffecter sans contrepartie une sous porteuse : dans ce cas l'utilisateur u_1 reçoit une sous porteuse provenant de l'utilisateur u_2 et l'utilisateur u_2 perd une sous porteuse.

Le processus décrit ci-dessus rentre le cadre de la décomposition 2 (allocation grossière + amélioration) illustrée en Fig. 3.

¹ A chaque sous porteuse de Ω_u , on associe la réduction de puissance qui serait acquise si la sous porteuse était affectée à v ; $\delta p_{u,v}$ est le maximum pris sur les réductions de puissance de toutes les sous porteuses de Ω_u .

Un autre processus (heuristique concernant uniquement l'allocation de sous porteuses) résout les collisions (cas où plusieurs utilisateurs ont la même meilleure sous porteuse) en attribuant des priorités entre les utilisateurs. C'est le cas dans [Pfletschinger&_02] : à chaque étape, l'utilisateur qui a la plus grande priorité choisit sa meilleure sous porteuse. Les priorités sont gérées de manière dynamique. Les priorités des utilisateurs sont mises à jour à chaque fois qu'un utilisateur reçoit une sous porteuse. La priorité initiale de chaque utilisateur est N_u/N . A chaque étape, la priorité de l'utilisateur u est la différence entre $N'_u/(\sum_u N'_u)$ et sa priorité initiale, avec N'_u le nombre de sous porteuses restant à allouer à l'utilisateur u .

4.2.2 Traitement par sous porteuses

On peut distinguer les algorithmes où les tâches 1 et 2 sont imbriquées (pas de détermination préalable du nombre de sous porteuses par utilisateurs) de ceux où ses tâches sont clairement séparées.

Pour décrire les algorithmes avec tâche 1 et 2 imbriquées, il est nécessaire de définir quelques notations. Ces algorithmes résolvent conjointement l'attribution de sous porteuses et les modulations transportées. Soit un utilisateur u auquel est alloué la sous porteuse n et sur laquelle il transporte $b_{u,n}$ bits. On définit $\Delta^+ P_{u,n}$ (respectivement $\Delta^- P_{u,n}$) comme l'incrément¹ de puissance pour transporter un² bit de plus (respectivement un bit de moins). On a les expressions suivantes :

- $\Delta^- P_{u,n} = P_{u,n}(b_{u,n}) - P_{u,n}(b_{u,n}-1)$
- $\Delta^+ P_{u,n} = P_{u,n}(b_{u,n}+1) - P_{u,n}(b_{u,n})$

$\Delta^- P_{u,n}$ représente l'économie en puissance réalisée par l'utilisateur u s'il transporte un bit de moins sur la sous porteuse n . $\Delta^+ P_{u,n}$ représente la dépense en puissance concédée par l'utilisateur u pour transporter un bit de plus sur la sous porteuse n .

4.2.2.1. Algorithmes avec tâches 1 et 2 imbriquées

Le processus décrit ci-après rentre dans le cadre de la décomposition 2 (allocation grossière + amélioration) illustrée en Fig. 3. Dans [Ergen_03], chaque sous porteuse (l'ordre de traitement est arbitraire) est attribuée à l'utilisateur qui peut transporter plus grand nombre de bits (ce qui initialise le nombre $b_{u,n}$ sur la sous porteuse). Le débit restant pour satisfaire l'utilisateur concerné est mis à jour. Une phase d'amélioration est réalisée afin de minimiser la puissance. Cette phase de minimisation comporte des échanges de sous porteuse (inspiré de [WongTsui_99]) et de modulation. Pour échanger les modulations entre 2 sous porteuses, un facteur de réduction de puissance spécifique est défini. L'algorithme identifie la puissance maximale $P_{u,n}(b_{u,n})$ sur tous les triplets (u,n,b) . Pour l'utilisateur u correspondant, $\Delta_n(u) = \max_{m \in \Omega_u} [\Delta^- P_{u,n} - \Delta^+ P_{u,m}]$, (cf. Fig. 7) ; si $\Delta_n(u) > 0$, on fait un « échange débit » entre les 2 sous porteuses (on retire un bit à $n = \arg \max P_{u,n}(b_{u,n})$ et on l'ajoute à $m^* = \arg \max_{m \in \Omega_u} [\Delta^- P_{u,n} - \Delta^+ P_{u,m}]$). L'algorithme s'arrête quand il n'y a plus d'améliorations possibles pour aucun utilisateur.

¹ Tels qu'ils sont définis, les 2 paramètres sont positifs.

² Nous décrivons ici les algorithmes tels qu'ils sont dans la littérature. Ils travaillent avec des nombres de bits $b_{u,n}$ entiers qui sont augmentés ou diminués d'un bit. Cependant, on pourrait étendre le raisonnement, à une table fixée de MCS, schémas de modulation et de codage. Le débit réel $r_{u,n}$ serait augmenté de α_{i+1} bits (ou diminué de α_i bits) ; nombres réels de bits nécessaires pour passer du $MCS(i)$ au $MCS(i+1)$ (ou du $MCS(i)$ au $MCS(i-1)$).

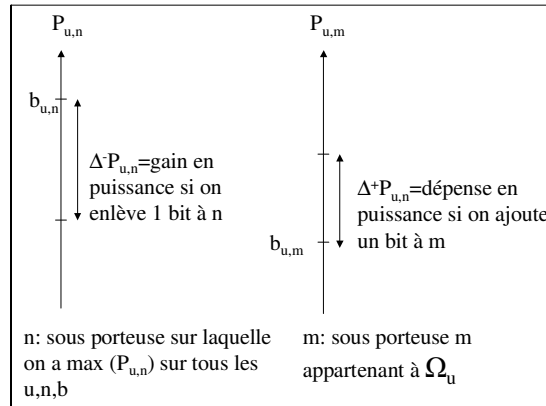


Figure 7: Echange de modulation entre 2 sous porteuses

Approche de [Zhang_04] : résolution de collisions

Dans l'initialisation de [Zhang_04], chaque utilisateur choisit ses sous porteuses comme s'il était le seul utilisateur. Pour chacun d'entre eux, le nombre de bit $b_{u,n}$ qu'il transporterait sur chacune des sous porteuse n est calculé par l'algorithme du *bit loading*¹. Une fois r_u° atteint, chaque utilisateur u possède alors une liste de sous porteuses désirées. Les sous porteuses apparaissant dans la liste d'un seul utilisateur sont attribuées à l'utilisateur correspondant. Les sous porteuses apparaissant dans les listes de plusieurs utilisateurs sont des sous porteuses « conflictuelles ». Elles sont traitées tour à tour (par ordre décroissant de la somme des puissances $P_{u,n}$ ($b_{u,n}$) des utilisateurs en conflit). Soit n une sous porteuse « conflictuelle », pour chaque utilisateur u en conflit, on calcule le coût en puissance $D_{u,n}$ s'il devait abandonner la sous porteuse n . Pour calculer $D_{u,n}$, l'auteur réaffecte les $b_{u,n}$ bits qui étaient affectés à n sur le reste des sous porteuses² grâce à l'algorithme du bit loading : la sous porteuse m qui minimise $\Delta^+P_{u,m}$ reçoit un bit de plus. Pour chaque u , $Dtot_{u,n} = \sum_{v \neq u} D_{v,n}$ représente le coût supplémentaire si u garde la sous porteuse n , l'utilisateur qui minimise cette dépense globale ($u^* = \arg \min_u Dtot_{u,n}$) reçoit la sous porteuse.

Approche de [Yu&_06] : allocation progressive du spectre

Dans l'initialisation de l'algorithme de [Yu&_06], chaque utilisateur u reçoit tour à tour (l'ordre est arbitraire) la meilleure sous porteuse (sur l'ensemble disponible) ; il y place son débit minimal de r_u° bits. Ensuite, à chaque nouvelle allocation, la sous porteuse n et l'utilisateur u auquel elle sera affectée sont déterminés conjointement. Il s'agit du couple (u,n) qui réalise la meilleure « *marginal utility* » : c'est l'économie de puissance réalisée par u s'il reçoit n comme sous porteuse supplémentaire. La réduction de puissance est calculée pour chaque utilisateur u en considérant qu'on lui ajoute sa meilleure sous porteuse (parmi les sous porteuses non allouées). La future répartition de bits sur Ω_u est effectuée comme suit : tant qu'on réduit la puissance de l'utilisateur u ($(\max_{m \in \Omega_u} \Delta^+P_{u,m}) > \Delta^+P_{u,n}$), on transfère un bit de la sous porteuse m ($\arg \max_{m \in \Omega_u} \Delta^+P_{u,m}$) à la sous porteuse candidate n . Dans cet algorithme, les conflits sont automatiquement réglés car si 2 utilisateurs désirent la même sous porteuse, c'est celui qui maximise la « *marginal utility* » qui la reçoit.

¹cf. 2.1.1 : jusqu'à ce que la somme des $b_{u,n}$ soit égale à r_u° , le débit minimal requis, la sous porteuse qui demande la plus faible augmentation de puissance pour transporter un bit de plus reçoit un bit supplémentaire. Cet algorithme est optimal pour minimiser la puissance transmise.

² On parle ici des sous porteuses de Ω_u . Les ex sous porteuses conflictuelles qui ont été attribuées à un utilisateur $v \neq u$ sont définitivement interdites pour l'utilisateur u .

4.2.2.2. Algorithmes avec tâches 1 et 2 séparées

L'approche de [KimKwak_04] utilise la notion de réduction de puissance. Les nombres N_u sont initialisés à $\lceil r_u^\circ / R_{max} \rceil$ (déjà évoqué en 3.1.1.3) pour la phase d'allocation initiale. Dans un ordre arbitraire, les usagers reçoivent une sous porteuse¹. Pour continuer la phase d'allocation spécifique de sous porteuses, le principe de réduction de puissance du BABS est utilisé. Le CgNR moyen $\bar{\varphi}_u$ est calculé sur les sous porteuses que possède l'utilisateur et non plus sur l'ensemble des sous porteuses. A chaque étape, on calcule $\Delta(u)$ (cf. 3.1.1) en considérant que la sous porteuse supplémentaire est la meilleure sous porteuse de l'utilisateur. Celui qui maximise $\Delta(u)$ se voit attribuer sa meilleure sous porteuse. Après l'allocation initiale, chaque utilisateur a $\lceil r_u^\circ / R_{max} \rceil$ sous porteuses. Chaque sous porteuse m restante va à celui qui maximise $\Delta(u)$ (calculé en considérant m comme la sous porteuse supplémentaire).

ACG et dérivés

[Kivanc&_00] propose l'algorithme ACG (*Amplitude Craving Greedy algorithm*). C'est un algorithme opportuniste dont le nombre de sous porteuses de chaque utilisateur est limité à N_u (fixé à la tâche 1). A chaque étape une sous porteuse n est fixée, l'algorithme alloue la sous porteuse n à l'utilisateur qui a le meilleur CgNR et qui n'a pas encore atteint son nombre de sous porteuses N_u . L'algorithme s'arrête lorsqu'il n'y a plus de sous porteuses à attribuer. La principale limitation de cet algorithme est l'ordre de traitement des sous porteuses. Elles sont traitées dans un ordre arbitraire ou dans un ordre aléatoire ([Kivanc&_03]). L'ordre de traitement a un impact² sur les performances.

Pour remédier au problème de l'ordre de traitement des sous porteuses, [Zhen&_03] propose un algorithme où la sous porteuse et l'utilisateur sont choisis conjointement de la façon suivante : $(u^*, n^*) = \arg \max_{u,n} \varphi_{u,n}$. Ainsi, à chaque étape on est assuré de réunir le couple qui réalise les meilleures performances. Dans la suite, l'algorithme proposé par [Zhen&_03]³ sera appelé « improvedACG ».

Une autre modification de l'ACG est exposée dans [Cho&_05], les auteurs ordonnent les sous porteuses selon le CgNR minimum, $\varphi_{min,n} = \min_{1 \leq u \leq U} \varphi_{u,n}$. On commence par les sous porteuses qui ont les $\varphi_{min,n}$ les plus faibles. Le principe d'affectation de l'ACG est conservé : allocation opportuniste en tenant compte de N_u . En affectant, très tôt, une sous porteuse de faible $\varphi_{min,n}$ à son meilleur utilisateur, on évite que le pire cas ne se produise. L'algorithme est appelé « modifiedACG ».

Les algorithmes ACG et dérivés décrits sont proposés dans un contexte MA, cependant s'ils sont précédés d'un algorithme tel que le BARE ils sont utilisables en RA. Nous testerons les performances des algorithmes ACG, « modifiedACG » et « improvedACG » dans la partie 5.

¹ Chaque utilisateur reçoit sa meilleure sous porteuse dans l'ensemble des sous porteuses disponibles.

² Prenons l'exemple d'un utilisateur u qui n'a besoin que d'une sous porteuse ($N_u=1$) et dont la meilleure sous porteuse est la dernière (index N). Il a très peu de chances de la recevoir. De plus, il n'y a aucune garantie sur les performances de la sous porteuse qu'il recevra à la place de sa meilleure sous porteuse.

³ [Zhen&_03] corrige une maladresse d'ACG en s'assurant qu'un usager qui a déjà reçu N_u sous porteuses soit retiré définitivement de l'ensemble des usagers susceptibles de recevoir des sous porteuses ; ceci est rendu possible par l'instruction $1 \leq n \leq N, \varphi(u,n) = -1$. L'ACG ne le fait pas et cela provoque des tests inutiles à chaque nouvelle sous porteuse. Prenons l'exemple d'un utilisateur pourvu dès la première sous porteuse (i.e. $N_u=1$) et meilleur utilisateur de toutes les autres. A chaque nouvelle sous porteuse, on teste si cet utilisateur est pourvu (i.e. $|\mathcal{Q}_u|=N_u$) ce qui fait $N-1$ tests inutiles car on aurait dû l'éliminer du processus d'allocation une fois son N_u satisfait.

4.3. Algorithmes heuristiques en RA

On distingue les articles qui résolvent un problème avec satisfaction de débits minimaux de ceux où il n'y a pas de débit minimal. On peut citer cependant [Anas_04], l'auteur se distingue en considérant simultanément deux classes d'utilisateurs GP (*guaranteed performances* i.e. débit minimal garanti) et BE (*best effort* i.e. pas de débit minimal).

4.3.1 Absence de contrainte débit minimal

Problème d'équité en débit

[RheeCioffi_00] tente de maximiser le débit le plus faible parmi les utilisateurs (en abrégé problème « max-min »). L'algorithme est initialisé en affectant à chaque utilisateur la sous porteuse sur laquelle il a le meilleur gain (l'ordre des utilisateurs est arbitraire). Les débits r_u sont initialisés en utilisant la formule de Shannon. Puis, tant qu'il reste des sous porteuses, l'utilisateur qui minimise le débit¹ r_u reçoit sa meilleure sous porteuse (parmi les sous porteuses disponibles). Cette étape est suivie de la mise à jour des débits r_u .

Un algorithme similaire est rencontré dans [Peng&_05], la métrique de l'utilisateur u n'est pas le débit r_u mais la somme des $\lg_{u,n} l^2$ sur Ω_u .

On peut noter que la phase concernant les utilisateurs BE (*best effort*) dans [Anas_04] est très largement inspirée par [RheeCioffi_00]. Cette phase est effectuée avec le nombre N' de sous porteuses disponibles après l'affectation aux utilisateurs GP (*guaranteed performance*).

Problème opportuniste

Les auteurs de [JanLee_03] et [KimHan&_04] tentent de maximiser le débit total sans imposer de contrainte de débit minimal. Ils montrent que la solution optimale est d'attribuer chaque sous porteuse à celui qui y présente le meilleur CgNR.

4.3.2 Optimisation RA avec la contrainte r°

Ce paragraphe passe en revue les différentes stratégies rencontrées pour l'affectation de sous porteuses quand les usagers doivent satisfaire un débit minimal.

Dans [Kivanc&_03], l'algorithme RCG (*Rate Craving Greedy algorithm*) attribue une sous porteuse à l'utilisateur qui y présente le meilleur débit estimé. Puis, des réaffectations sont réalisées : les utilisateurs dont $|\Omega_u|$ dépasse N_u (fixés par la tâche 1) sont parcourus. Soit v un tel utilisateur, tant que v a un excédent de sous porteuses, on choisit parmi les utilisateurs u qui ont $|\Omega_u| < N_u$, celui qui a le meilleur débit estimé sur Ω_v^2 . La meilleure sous porteuse de l'utilisateur u correspondant est retirée de Ω_v et affectée à l'utilisateur u^3 .

Les auteurs de [Gross&_03] choisissent d'attribuer la totalité des N_u meilleures sous porteuses de chaque usager u quand celui-ci est pris en compte dans le processus d'affectation. L'algorithme est appelé bDA (*basic Dynamic Assignment*). Les auteurs font évoluer les priorités de façon cyclique. Tous les S symboles, chaque utilisateur voit sa priorité s'affaiblir excepté la priorité la plus faible qui devient la priorité la plus forte. L'équité est donc réalisée sur U symboles.

¹ Dans la suite, on appelle le critère à minimiser : la métrique.

² D'après l'article, le meilleur débit estimé est pris sur l'ensemble des sous porteuses, il serait plus logique que ce soit sur l'ensemble $\in \Omega_v$. L'objet de la boucle est en effet à réduire le cardinal de cet ensemble.

³ Dans l'article, la meilleure sous porteuse de u est prise sur l'ensemble des sous porteuses mais ce ne serait pas logique : comment retirer à $|\Omega_v|$ une sous porteuse qui ne lui appartient pas ? Nous pensons donc que « arg min $(r_v(n) - r_u(n))$ » avec $1 < n < N$ est une erreur de frappe, il faudrait plutôt $n \in \Omega_v$.

Dans [Anas_04], les utilisateurs GP (*guaranteed performance*) choisissent tour à tour leur meilleure sous porteuses jusqu'à satisfaction de leur débit minimal. Le nombre de sous porteuses N' restant est utilisé pour les utilisateurs de classe BE.

Approche de [Wong&_04]

Dans [Wong&_04], le vecteur de débit final $\underline{r}$ doit vérifier pour tout couple d'utilisateurs (u,v) $r_u/r_v = \beta_u/\beta_v$ où $\underline{\beta}$ est donné en entrée. On peut parler de *fair proportionnal constraints*. Si le vecteur $\underline{\beta}$ est le vecteur unité (i.e. $\underline{1}$), on obtient un problème similaire au « max-min ».

Trois phases constituent l'algorithme de [Wong&_04].

Le nombre de sous porteuses par usager est initialisé à $N_u = \lfloor \beta_u \cdot N \rfloor$. Pendant l'initialisation (première phase), les U usagers (suivant un ordre arbitraire) reçoivent leur meilleure sous porteuse (dans l'ensemble des sous porteuses disponibles).

La deuxième phase s'arrête lorsque chacun possède N_u sous porteuses. Le principe d'attribution ressemble à celui de [RheeCioffi_00] : l'utilisateur qui minimise la métrique choisit sa meilleure sous porteuse (meilleur CgNR dans l'ensemble des sous porteuses disponibles). La métrique diffère de [RheeCioffi_00] (pour $\underline{\beta} \neq \underline{1}$) : elle vaut r_u/β_u .

Quand $\sum_{u=1}^U N_u \leq N$, il reste des sous porteuses non allouées à la fin de la 2^{ème} phase. La dernière phase alloue chacune des ces sous porteuses à qui y présente le meilleur gain (inspiré par le théorème de [JanLee_03]).

On peut noter que [HuiZhou_06] propose un algorithme très largement inspiré de [Wong&_04]. La seule différence est l'introduction de priorités entre les utilisateurs au moment de la première phase (pour pallier un ordre arbitraire). Les 2 dernières phases sont inchangées.

4.4. Un nouvel algorithme : le RPO

Pour la réalisation de la tâche 2, nous proposons le RPO (*Rate Profit Optimization* [Lengoumbi&_06]) qui est une nouvelle alternative à l'algorithme Hongrois et aux heuristiques décrites dans les sections précédentes.

4.4.1 Motivations

Nous voulons disposer d'un algorithme qui ne défavorise pas certains utilisateurs à cause d'un ordre de traitement arbitraire. Il arrive souvent que des utilisateurs différents aient la même meilleure sous porteuse. Comment résoudre le conflit ? Dans la plupart des algorithmes rencontrés, le conflit est résolu par des priorités arbitraires entre les utilisateurs (sans garantie sur les conséquences sur le débit total). D'autres fois, l'utilisateur qui a le meilleur CgNR reçoit la sous porteuse. Cela est parfois injuste car on peut priver un utilisateur d'une unique très bonne sous porteuse au profit d'un utilisateur qui a plusieurs sous porteuses de très bonne qualité. L'algorithme RPO propose une règle de résolution de conflit plus juste par rapport aux utilisateurs et plus profitable au niveau du débit global.

4.4.2 Description

Un utilisateur satisfait est un utilisateur qui a reçu N_u sous porteuses. Dans ce cas, il n'est plus en compétition pour recevoir des sous porteuses. L'algorithme se termine lorsque toutes les sous porteuses sont affectées ($A=\emptyset$). A chaque itération, la meilleure sous porteuse de chaque utilisateur (non satisfait) est déterminée.

Soit C l'ensemble des sous porteuses désignées par plus d'un utilisateur. Les sous porteuses n'appartenant pas à C sont allouées à l'utilisateur qui les a désigné (il n'y a pas de conflit). Lors d'une itération avec $C=\emptyset$, $|I|$ sous porteuses sont allouées où I est l'ensemble des utilisateurs non satisfaits. Le nombre d'itérations du RPO est situé entre N/U et N . Lorsque l'ensemble C n'est pas vide, chaque sous porteuse de C (l'ordre de traitement n'a aucune importance) est allouée à l'utilisateur présentant le meilleur profit.

Pourquoi la notion de profit ?

Dans les algorithmes à tendance opportuniste (ACG et dérivés), chaque sous porteuse n est attribuée à son meilleur utilisateur sans en vérifier les conséquences sur le débit global. Lorsqu'il y a une collision sur n , nous évaluons l'impact, sur le débit global, de l'attribution de cette sous porteuse aux différents utilisateurs possibles. Le profit de l'utilisateur u représente le gain du débit global si u reçoit la sous porteuse n à la place du meilleur utilisateur de n .

Calcul du profit

Soit $n \in C$, A_n est l'ensemble des utilisateurs qui ont désigné n comme meilleure porteuse. Seuls les utilisateurs de A_n sont susceptibles de recevoir n . Soit $u^\circ(n)$ le meilleur d'entre eux du point de vue du CgNR ; les algorithmes ACG et dérivés alloueraient n à $u^\circ(n)$ ¹.

La seconde meilleure sous porteuse $n''(u)$ de chaque utilisateur u de A_n est déterminée. En effet, à une étape donnée, si un utilisateur u ne reçoit pas sa meilleure sous porteuse², il postulera pour sa seconde meilleure sous porteuse à l'étape suivante. On appelle $rateGap(u)$ le débit perdu par l'utilisateur u s'il on ne lui attribue pas sa meilleure sous porteuse. Alors $rateGap(u)$ ³ est l'écart entre le débit de la meilleure sous porteuse $n'(u)$ de u et le débit de u sur sa seconde meilleure sous porteuse $n''(u)$.

Si on attribue la sous porteuse n à u plutôt qu'à u° , l'utilisateur u gagne $rateGap(u)$ bits et l'utilisateur u° perd $rateGap(u^\circ)$ bits. Finalement, si on attribue la sous porteuse n à u plutôt qu'à u° , le débit global gagne $rateGap(u) - rateGap(u^\circ)$.

Le profit de l'utilisateur u est donc définie comme suit : $profit(u) = rateGap(u) - rateGap(u^\circ)$. Le profit de l'utilisateur $u^\circ(n)$ est par définition nul. On veut améliorer le débit global : un utilisateur de A_n et différent de u° ne reçoit la sous porteuses n que s'il a un profit positif. Si plusieurs d'entre eux ont un profit positif, celui qui maximise le profit est choisi. L'utilisateur $u^\circ(n)$ reçoit la sous porteuse si tous les autres ont un profit négatif.

Tableau 2: Notations de l'algorithme RPO

$n'(u)$	$\arg \max_n \varphi_{u,n}$ i.e. meilleure sous porteuse de l'utilisateur u
$n''(u)$	$\arg \max_{n \neq n'(u)} \varphi_{u,n}$ i.e. seconde meilleure sous porteuse de l'utilisateur u
$u^\circ(n)$	$\arg \max_u \varphi_{u,n} (u \in A_n)$
A	Ensemble des sous porteuses non allouées
C	Ensemble des sous porteuses présentant un conflit
Ω_u	Ensemble des sous porteuses de l'utilisateur u
I	Utilisateurs insatisfaits ($ I < N_u$)
A_n	Utilisateurs dont n est la meilleure sous porteuse $n'(u)=n$; ($n \in C$)

¹ La notation est allégée dans la suite et devient u° quand n est fixée.

² La meilleure sous porteuse d'un utilisateur u quelconque est notée $n'(u)$. On a $n'(u) = n$ pour tous les utilisateurs de A_n .

³ L'expression de $rateGap(u)$ est $h(\gamma_{u,n'(u)}) - h(\gamma_{u,n''(u)})$ et représente l'écart de performance entre les 2 meilleures sous porteuses de u .

RPO Algorithm

$A = \{1..N\}$ # all subcarriers

$\Pi = \{1..U\}$ # all users

while $|A| > 1$

 for each $u \in \Pi$

$n'(u) = \arg \max_n (\varphi_{u,n})$;

$n''(u) = \arg \max_{n \neq n'(u)} (\varphi_{u,n})$;

 end for

Let C be the set of subcarrier such that $n'(u) = n'(v)$ and $u \neq v$;

 for each $u \in \Pi$

 # Simple cases resolution

 if $n'(u) \notin C$,

$\mathcal{Q}_u = \mathcal{Q}_u \cup \{n'(u)\}$;

$A = A - \{n'(u)\}$;

 if $|\mathcal{Q}_u| = N_u$,

$\Pi = \Pi - \{u\}$;

 end if

 end if

 end for

 while $|C| \neq 0$, pick n in C # Collision cases resolution

$u^\circ = \arg \max_{u \in \Pi} (\varphi_{u,n})$;

$r_{u^\circ,1} = h(\gamma_{u^\circ,n})$;

$r_{u^\circ,2} = h(\gamma_{u^\circ,n''(u)})$;

$rateGap(u^\circ) = r_{u^\circ,1} - r_{u^\circ,2}$;

A_n is formed;

 for each $u \in A_n$

$r_{u,1} = h(\gamma_{u,n})$;

$r_{u,2} = h(\gamma_{u,n''(u)})$;

$rateGap(u) = r_{u,1} - r_{u,2}$;

$profit(u) = rateGap(u) - rateGap(u^\circ)$;

 end for

$u' = \arg \max_{u \in A_n} (profit(u))$;

$\mathcal{Q}_{u'} = \mathcal{Q}_{u'} \cup \{n\}$;

 if $|\mathcal{Q}_{u'}| = N_{u'}$,

$\Pi = \Pi - \{u'\}$;

 end if

$C = C - \{n\}$;

$A = A - \{n\}$;

 end while

end while

The last subcarrier (if any), is assigned to the last user of Π .

5. Simulation et résultats

5.1. Motivations

Dans cette partie, nous résolvons le problème RA avec une équité en débit minimal :

Maximiser $\sum_u r_u$

avec $P_T < P_{T,Max}$, (1)

et pour tout u , $r_u \geq r^o$ (2a)

Pour cela, nous proposons le BARE pour la détermination du nombre de sous porteuses par utilisateurs (tâche 1) et le RPO pour l'affectation spécifique de sous porteuses (tâche 2). Les objectifs de cette partie sont les suivants :

- comparer les algorithmes BARE et BABS pour la tâche 1,
- comparer les performances de divers algorithmes de la tâche 2.

Les algorithmes de tâche 2 en « compétition » sont le RPO, l'algorithme Hongrois, les algorithmes ACG, « improvedACG », « modifiedACG » et le bDA. Les nombres N_u en entrée de tous ces algorithmes sont donnés par le BARE. La fonction de coût choisie pour l'algorithme Hongrois est $-\log_2(1 + p \varphi_{u,n})^1$. Nous nous intéressons aux performances suivantes : débit moyen par sous porteuses, facteur d'équité, probabilité d'*outage* par rapport au débit minimal commun et complexité. Nous nous intéressons aussi aux performances de l'algorithme opportuniste ([JanLee_03]) et de l'heuristique équitable ([RheeCioffi_00], problème « max-min »).

5.2. Modèle canal et Paramètres du système

Nous considérons une cellule constituée d'une station de base et de U utilisateurs uniformément répartis. Le modèle canal adopté est constitué de N sous porteuses parallèles sur la bande B . Aucune corrélation n'est considérée entre les sous porteuses. Le gain du canal de chaque sous porteuse n par rapport à l'utilisateur u est : $|g_{u,n}|^2 = K d(u)^{-\alpha} a_{sh}(u) a_f(u,n)$. Dans cette formule, K est l'atténuation à la distance de référence 1 m ; $d(u)$ est la distance de l'utilisateur ; l'exposant α du modèle d'atténuation régit la rapidité de la décroissance du signal en fonction de la distance ; $a_{sh}(u)$ est une variable lognormale qui traduit l'effet de masque subi par toutes les sous porteuses de l'utilisateur u ; enfin, $a_f(u,n)$ est une variable de Rayleigh qui traduit l'évanouissement rapide (*fast fading*) de la sous porteuse n .

Les sous porteuses subissent par ailleurs un bruit blanc gaussien de variance $\sigma^2 = N_0 B/N$. Ainsi le CgNR (*Channel gain to Noise Ratio*) est donné par $\varphi_{u,n} = |g_{u,n}|^2 / \sigma^2$ et le SNR (*Signal to Noise Ratio*) s'en déduit simplement : $\gamma_{u,n} = P_{u,n} \varphi_{u,n}$. Rappelons que nous adoptons une allocation de puissance uniforme $P_{u,n} = p$ sur chaque sous porteuse ([RheeCioffi_00], [JanLee_03], [Gross&_03], [Anas_04], [HuiZhou_06]). Pour calculer le débit à partir du SNR, nous utilisons comme fonction $h(\gamma) = \min(R_{max}, f^{-1}(\gamma))$ avec la fonction $f(\cdot)$ proposée par [Kivanc&_03]. La puissance p par sous porteuses varie de 1 à 35 mW (0 à 15 dBm). Pour $N=128$, la puissance émise par la BS varie alors de 21 à 36 dBm.

¹ La fonction de coût est exprimée sans approximation sur la puissance, car nous connaissons à l'avance l'allocation de puissance sur les sous porteuses : $p = P_{T,max}/N$.

Tableau 3: Paramètres de simulation

B (MHz)	1
U	8
N	128
λ (bit)	1
R_{max} (bits)	4.5
K : Path loss constant	10^{-4}
α : Path loss exponent	2.8
D_{max} (km)	5
σ_{sh} (dB)	8
Mean of exponential variable : a_f^2	1
Standard deviation of a_f^2	1
Power p per subcarrier (mW)	1...35
N_0 Thermal noise density (dBm/Hz)	-174

5.3. Débit par sous porteuse

5.3.1 Comparaison des algorithmes BABS et BARE

L'algorithme BARE (*Bandwidth Allocation on Rate Estimation*) a été introduit afin de disposer d'un algorithme mieux adapté au cas RA (utilisant les N sous porteuses et le budget $P_{T,max}$ disponible). Pour comparer le BARE et le BABS, il suffit d'utiliser les ensembles $\{N_u\}_{1 \leq u \leq U}$ qu'ils construisent à l'entrée de l'algorithme Hongrois. L'algorithme Hongrois apporte la solution optimale à la tâche 2 pour un ensemble $\{N_u\}_{1 \leq u \leq U}$ fixé. Le meilleur algorithme de la tâche 1 peut être désigné en comparant le débit par sous porteuses obtenu en utilisant les algorithmes BARE et BABS avant l'algorithme Hongrois.

Le BABS est utilisé avec les paramètres suivants :

- initialisation des N_u à $\lceil r^\circ / R_{max} \rceil^1$
- condition d'arrêt : nombre total N de sous porteuses allouées

Dans le cadre de l'équité en débit minimal, la boucle mise en œuvre pour le BABS est :

```
While  $P > P_{max}$ 
     $r^\circ = r^\circ - \lambda$ 
    BABS
```

end

Dans le cadre de l'équité en débit minimal (cf. 3.2.3.1), la boucle mise en œuvre pour le BARE est :

```
While  $(\min_u (gap(u, N_u)) < 0) \& (r^\circ > \lambda)$ 
     $r^\circ = r^\circ - \lambda$ 
    BARE
```

end

Si une contrainte est trop élevée, le problème se pose en terme de dépassement du budget de puissance dans le BABS, tandis que dans le BARE il se pose en terme d'*outage* (nombre d'utilisateurs insatisfaits).

La figure 9 montre que le BARE permet d'obtenir un meilleur débit par sous porteuse, la répartition $\{N_u\}_{1 \leq u \leq U}$ qu'il propose est donc plus efficace que celle du BABS. La différence se situe entre 0.12 et 0.34 bit/s/Hz. Dans la suite du document, l'algorithme BARE est adopté pour exécuter la tâche 1.

¹ Lors de l'initialisation, si $U \lceil r^\circ / R_{max} \rceil > N$, il est inutile de continuer la tâche : il n'y a pas de solution et on choisit de recommencer avec $r^\circ = r^\circ - \lambda$.

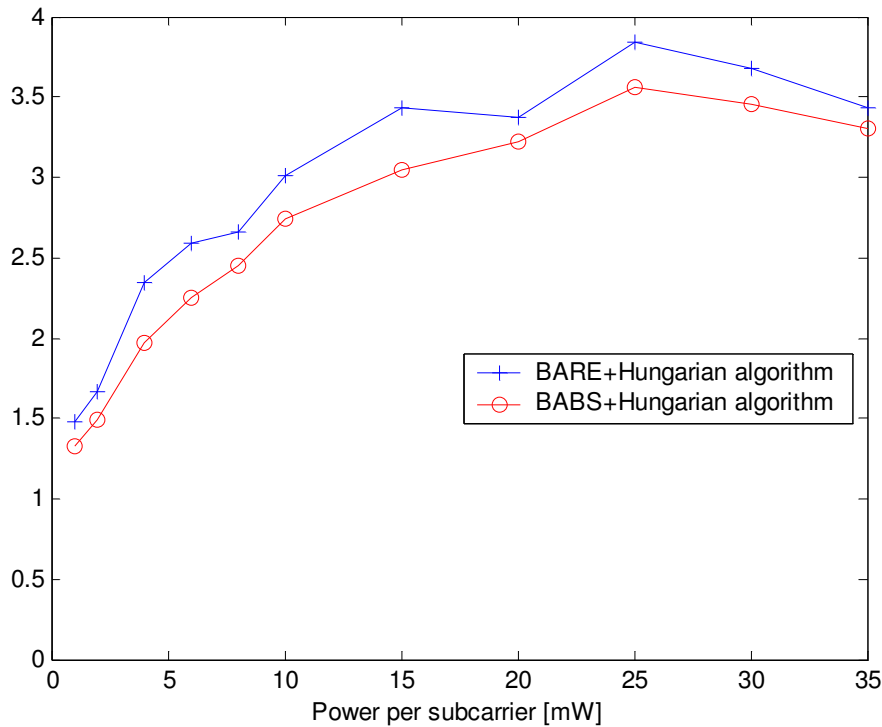


Figure 8: Comparaison de BABS et BARE (débit moyen par sous porteuses)

5.3.2 Comparaison des algorithmes de la tâche 2

Impact des stratégies d'équité

Garantir à chaque utilisateur u un débit minimal r_u° ou maximiser le débit minimal d'un utilisateur provoque une baisse de la capacité globale par rapport à une stratégie purement opportuniste. La réduction sur la capacité globale provoquée par la garantie d'un débit minimal correspond à la différence de débit entre l'algorithme opportuniste et l'algorithme Hongrois (précédé du BARE). La réduction due à l'équité totale en débit (stratégie « max-min ») correspond à la différence de débit entre l'algorithme opportuniste et l'heuristique équitable ([RheeCioffi_00]).

Nous avons cherché à garantir un débit minimal r° commun à chaque utilisateur. Nous qualifions cela d'équité en débit minimal. Sur la figure 9, on voit que les stratégies d'équité pénalisent moins le débit lorsque la puissance augmente. La réduction due à l'équité en débit minimal diminue de 30% à 4% tandis que celle due à l'équité totale diminue de 46% à 7% quand p varie de 1 à 15 mW par sous porteuses.

Comparaison : algorithme Hongrois, RPO, ACG, « modifiedACG », « improvedACG »

L'algorithme Hongrois donne la solution optimale par rapport à la répartition $\{N_u\}_{1 \leq u \leq U}$ indiquée par le BARE. Il présente donc la meilleure efficacité spectrale. L'algorithme RPO présente de bonnes performances car il talonne l'algorithme Hongrois de près (moins d'1% de différence). La différence du RPO avec l'algorithme Hongrois est stable : elle ne dépend ni de la puissance ni des paramètres du modèle (K, α, σ_{sh} , etc...). Le bDA et l'algorithme « improvedACG » suivent avec respectivement 2% et 3% de différence. Les performances de l'algorithme « modifiedACG » sont assez variables et dépendent beaucoup des paramètres du

modèle de propagation. L'écart entre cet algorithme et l'algorithme Hongrois baisse de 8% à 3% avec la puissance. A partir de $p=10\text{mW}$, il surpasse « improvedACG ». On peut remarquer cependant que lorsque $\alpha > 3$, l'efficacité de « modifiedACG » est toujours inférieure à celle de « improvedACG ». On peut espérer d'un algorithme qu'il soit insensible aux paramètres du modèle : ce n'est pas le cas de « modifiedACG » du point de vue du débit. L'algorithme ACG présente les performances les plus faibles à cause de l'impact de l'ordre de traitement des sous porteuses.

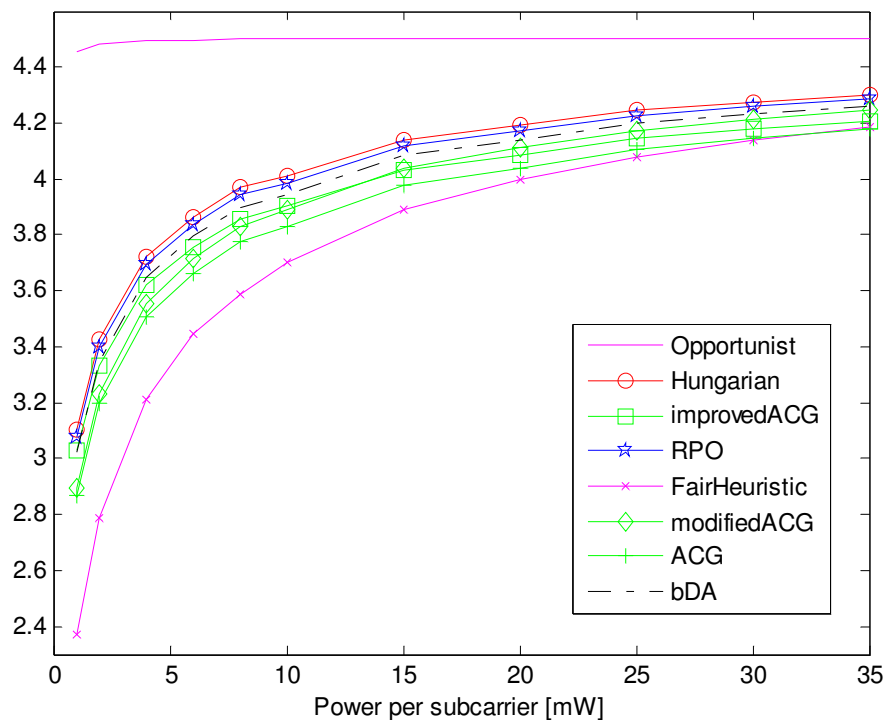


Figure 9: Débit moyen par sous porteuse

5.4. Facteur d'équité en débit entre les utilisateurs

Dans le cadre de l'équité en débit minimal, on définit un facteur d'équité noté F qui est le rapport entre le débit le plus faible et le débit le plus élevé rencontrés dans la cellule : $F = \min_u r_u / \max_u r_u$

Une équité parfaite est caractérisée par un facteur F de 1. L'équité de l'heuristique de [RheeCioffi_00] (stratégie « max-min ») est confirmée par un facteur de 0.9 tandis que « l'injustice » de l'algorithme opportuniste est souligné par facteur F nul. L'algorithme « modifiedACG » traite les mauvaises sous porteuses en premier afin qu'elles ne soient pas allouées à leur pire utilisateur, cet ordre de traitement permet à l'algorithme de réaliser un facteur d'équité très proche de l'algorithme Hongrois. Le RPO s'assure qu'un utilisateur ne perde pas sa meilleure sous porteuse lorsque sa seconde meilleure sous porteuse est mauvaise. Cette règle assure au RPO un facteur F proche de ceux de l'algorithme Hongrois et du « modifiedACG ». Le RPO surpasse ainsi les algorithmes bDA, ACG et « improvedACG ».

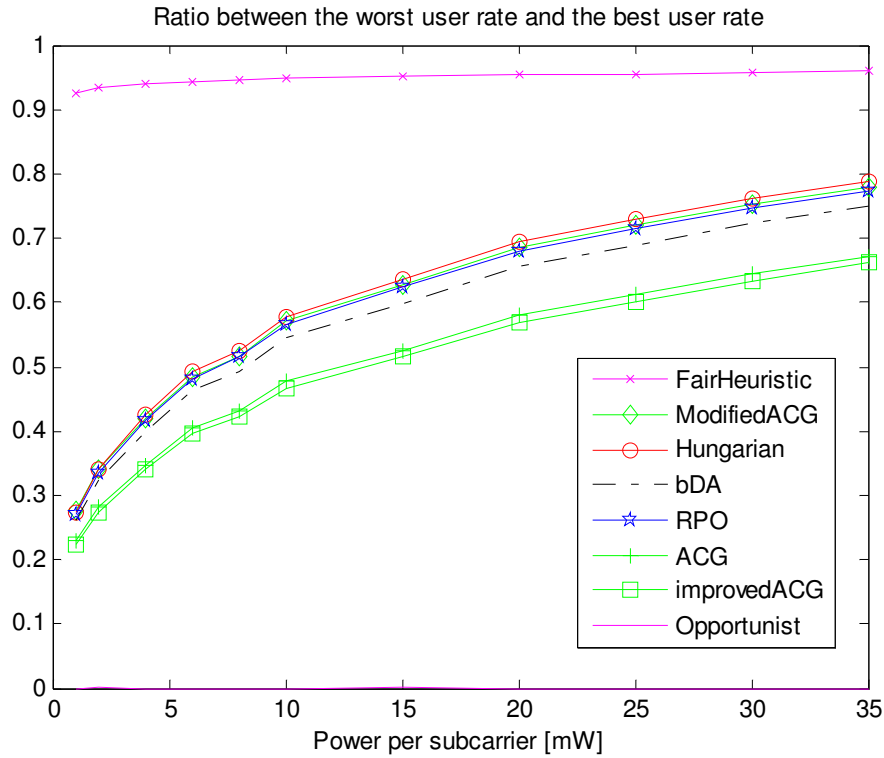


Figure 10: Facteur d'équité, $F = \min_u r_u / \max_u r_u$

5.5. Probabilité d'échec de la tâche 2

La figure 11 donne l'évolution du débit minimal commun r° (évalué par la tâche 1) avec la puissance, il augmente entre 10 et 25 bits par symboles OFDM. Les algorithmes utilisés pour la tâche 2 doivent assurer ce débit minimal car le nombre de sous porteuses N_u par utilisateur u a été calculé dans ce but. On veut évaluer la probabilité qu'un utilisateur n'atteigne pas le débit minimal commun r° . Nous définissons ce phénomène comme un *outage*.

L'algorithme Hongrois n'apparaît pas Fig. 12 car sur 20000 itérations pour chaque valeur de la puissance p , aucun *outage* n'a été recensé. La probabilité d'*outage* de l'algorithme Hongrois est donc inférieure à 10^{-6} . Un faible *outage* de l'algorithme Hongrois valide les

répartitions $\{N_u\}_{1 \leq u \leq U}$ obtenues par le BARE. L'algorithme Hongrois donne en effet la solution optimale correspondant à une répartition $\{N_u\}_{1 \leq u \leq U}$ fixée. Si l'*outage* de l'algorithme Hongrois est élevé, on peut en déduire que la répartition $\{N_u\}_{1 \leq u \leq U}$ d'entrée est « mauvaise ».

Sur un total de 200 000 itérations, seuls deux cas d'*outage* ont été recensés pour l'algorithme RPO ; ceci donne un ordre de grandeur de 10^{-5} pour la probabilité d' *outage* de l'algorithme RPO. Il surpasse les algorithmes « modifiedACG » (10^{-4}), bDA (10^{-3}) ainsi que les algorithmes ACG et « improvedACG » qui ferment la marche avec des probabilités d' *outage* de 10^{-2} .

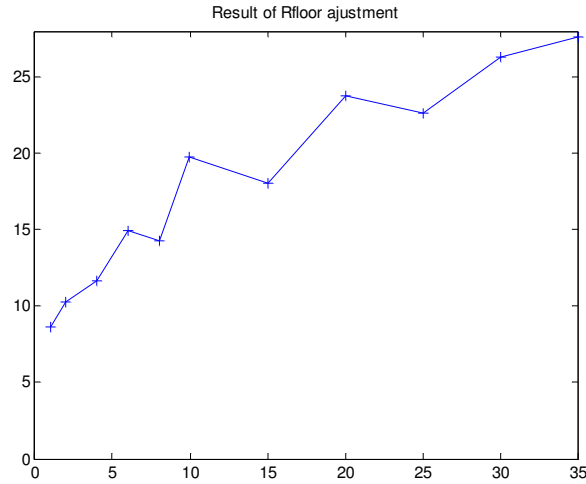


Figure 11: Débit minimal commun à la fin de la tâche 1 (exprimé en nombre de bits par symboles OFDM) en fonction de la puissance par sous porteuses en mW

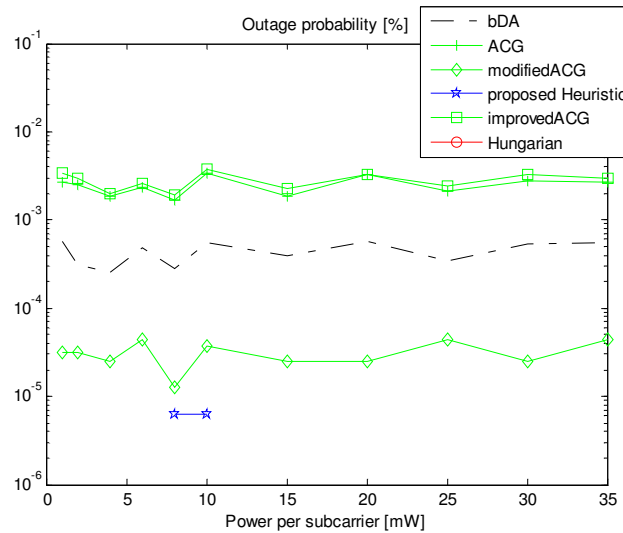


Figure 12: Probabilité d'échec

5.6. Complexité des algorithmes de la tâche 2

La complexité des algorithmes est un paramètre important pour analyser leur « faisabilité ». Le temps de cohérence du canal est très court (de l'ordre de la milliseconde), l'allocation doit donc être assez rapide pour que le résultat soit valide.

On trouve pour l'algorithme Hongrois deux complexités dans la littérature $\mathcal{O}(N^3)$ ([Gross&_03], [DraHen_04]) et $\mathcal{O}(N^4)$ ([WongTsui_99], [Chen&_04]). Il n'est pas précisé s'il s'agit des complexités en moyenne ou au pire cas. Dans tous les cas, la complexité de l'algorithme n'est pas envisageable lorsque N augmente. C'est pourquoi, des algorithmes sous optimaux sont proposés. La complexité de l'ACG est donnée en $\mathcal{O}(N \times U)$ ([Kivanc&_00]). [Zhen&_03] ne fournit pas la complexité de l'algorithme « improvedACG » mais on peut l'évaluer. A chaque itération, U utilisateurs choisissent leur meilleure sous porteuses parmi N puis on choisit le meilleur utilisateur parmi U . Chaque itération est donc en $\mathcal{O}(N+U)$. Il y a N itérations donc la complexité de l'« improvedACG » est en $\mathcal{O}(N^2)$.

L'heuristique proposée commence par trouver pour chacune des N sous porteuses le meilleur utilisateur parmi U . Puis les N sous porteuses sont classées en ordre décroissant : cette opération est en $\mathcal{O}(N \log N)$. La complexité de l'algorithme est $\mathcal{O}(U \times N + N \log N)$.

Lorsqu'on fait varier N entre 16 et 2048 et que l'on observe le temps d'exécution de l'allocation pour 4 utilisateurs, on obtient la figure 13. L'algorithme opportuniste est le plus rapide car le plus simple. Les algorithmes heuristiques sont jusqu'à mille fois plus rapide que l'algorithme Hongrois. On voit que l'heuristique proposée rivalise en temps d'exécution avec l'ACG, il est donc meilleur que l'algorithme « improvedACG » en terme de rapidité ; on a vu dans la section précédente que ses performances rivalisaient avec l'« improvedACG ». L'heuristique proposée réunit donc la rapidité de l'ACG et les performances de l'« improvedACG ».

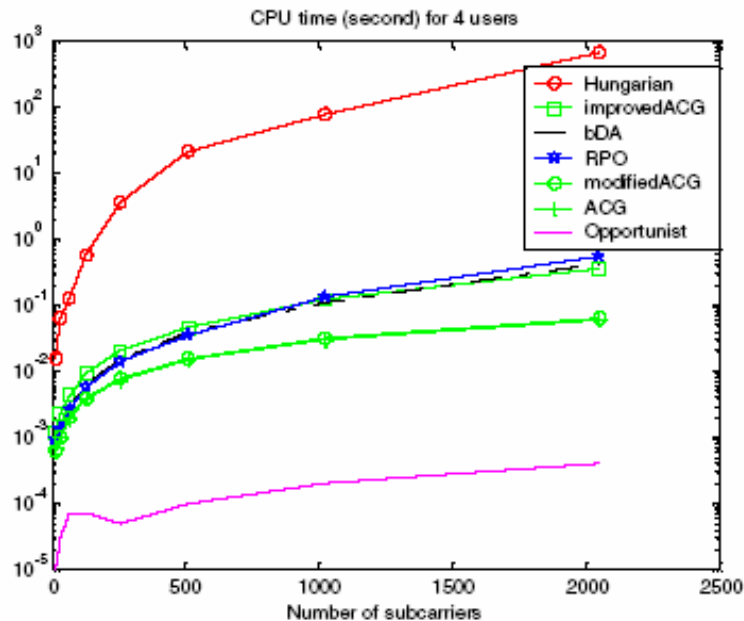


Figure 13: Durée d'exécution des algorithmes (tâche 2) pour 4 utilisateurs

6. Conclusion

Nous avons réalisé une étude bibliographique sur l'allocation de ressources en OFDMA. Deux principaux problèmes sont traités dans la littérature MA et RA. Le problème MA minimise la puissance totale transmise tandis que le problème RA maximise le débit global de la cellule. Dans ces deux types de problèmes, l'optimisation est simplifiée lorsqu'on distingue l'attribution de sous porteuses et l'allocation de puissance (ou le choix des MCS). L'attribution de sous porteuses est souvent réalisée en deux tâches successives : l'allocation de largeur de bande et l'affectation spécifique des sous porteuses. Ces tâches, respectivement appelées tâche 1 et tâche 2, sont aussi bien considérées en MA qu'en RA.

Dans les problèmes MA, certains auteurs réalisent l'attribution de sous porteuses et le choix des MCS de façon conjointe. Dans ce cas, ils considèrent une phase d'allocation grossière et une phase d'amélioration. Dans la phase d'amélioration, pour diminuer la puissance transmise, les sous porteuses peuvent être échangées entre deux utilisateurs distincts et les MCS peuvent être « échangés » sur les sous porteuses d'un utilisateur.

Nous avons proposé des algorithmes pour la résolution du problème RA. Nous avons utilisé lors de cette résolution :

- les tâches 1 et 2 lors de l'attribution des sous porteuses,
- une allocation uniforme de puissance sur les sous porteuses car une étude montre que les performances rivalisent avec le *waterfilling* ([JanLee_03]).

Concernant le calcul du nombre de sous porteuses par utilisateurs (allocation de largeur de bande, tâche 1), nous avons proposé le BARE (*Bandwidth Allocation on Rate Estimation*) qui, nous l'avons montré, est plus efficace dans le contexte RA que le BABS (*Bandwidth assignment based on SNR*). Au cours de cette tâche, nous avons cherché à assurer une équité en débit minimal. Néanmoins, le BARE est également adapté à une contrainte quelconque sur le vecteur de débit.

Concernant l'affectation spécifique de sous porteuses (tâche 2), nous proposons le RPO (*Rate Profit Optimization*). Son but est d'éviter l'allocation « aveugle » d'une sous porteuse à son meilleur utilisateur. Lorsque plusieurs utilisateurs ont une meilleure sous porteuse commune, la seconde meilleure sous porteuse de chacun est évaluée. Ainsi le débit global n'est pas pénalisé par une affectation systématique au meilleur utilisateur de la sous porteuse considérée. Pour comparer les performances des heuristiques, on utilise l'algorithme Hongrois qui est l'algorithme optimal pour la tâche 2 (la fonction de coût considérée pour ce problème RA est $c(u,n) = -\log_2(1+p\phi_{u,n})$).

D'après les simulations, le RPO réalise l'efficacité spectrale la plus proche de l'algorithme Hongrois. Le facteur d'équité du RPO est de même très proche de celui de l'algorithme Hongrois. La probabilité d'échec du RPO, c'est-à-dire la probabilité qu'un utilisateur n'atteigne pas le débit minimal, surpasse celle de toutes les heuristiques rencontrées. Enfin, la complexité du RPO est comparable aux heuristiques existantes. Ainsi, pour une complexité comparable, le RPO surpasse les heuristiques existantes au regard de l'efficacité spectrale, du facteur d'équité et de la probabilité d'échec de la tâche 2 (par rapport aux objectifs définis par la tâche 1). Le RPO est donc une bonne alternative à l'algorithme Hongrois.

Annexe A : Puissance démission et *SNR gap*

La puissance nécessaire pour transmettre un débit r sur la sous porteuse n est :

$$P(r) = \frac{\Gamma(2^r - 1)}{\varphi} \quad (1)$$

Dans cette formule, φ est le CgNR (Channel Gain to Noise Ratio); il est égal au rapport entre le gain g du canal (résultat de l'atténuation, de l'effet de masque) et le bruit présent sur la bande d'une sous porteuse (Δf) : $\varphi = \frac{|g|^2}{N_0 \Delta f}$ (2), ([YinLiu_00],[Wong&_04]).

Le paramètre Γ (connu sous le nom de SNR-gap où SNR signifie Signal Noise To Ratio) est la marge de puissance qu'il faut concéder pour assurer un SER (Symbol Error Rate) donné.

Expression de Γ

Exprimons la probabilité d'erreur symbole d'une M-QAM à $|g|$ fixé (où $|g|$ suit une loi de rayleigh) ([Iamy_00, p126]):

$$SER \approx 4Q\left(\sqrt{\frac{3 \log_2(M) |g|^2 E_b}{(M-1) N_0}}\right) \quad (3) \text{ avec } Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$$

Cette formule de la probabilité d'erreur d'une M-QAM s'établit en considérant une M-QAM comme $2 \sqrt{M}$ -PAM (Pulse Amplitude Modulation) indépendantes en quadrature [p231, benedetto_82].

On a $E_b = P(r) T_b$ où la puissance d'émission d'un symbole QAM de $r = \log_2(M)$ bits et $T_b = \frac{1}{\log_2(M) \Delta f}$. Dans (3) nous remplaçons donc E_b par $\frac{P(r)}{\log_2(M) \Delta f}$ et M par 2^r . D'où

$$SER \approx 4Q\left(\frac{3P(r)|g|^2}{(2^r - 1) N_0 \Delta f}\right) \quad (4).$$

d'après (2) $SER \approx 4Q\left(\frac{3P(r)\varphi}{(2^r - 1)}\right)$. On en déduit que $P(r) \approx \frac{2^r - 1}{\varphi} \frac{Q^{-1}\left(\frac{SER}{4}\right)}{3}$ (4).

Par identification entre (1) et (4), on obtient que $\Gamma = \frac{Q^{-1}\left(\frac{SER}{4}\right)}{3}$ ([MunzPS_02]).

Annexe B : L'algorithme Hongrois

Le problème qui se pose est l'affectation de chaque sous porteuse à un utilisateur unique. La liaison entre un utilisateur et une sous porteuse est associée à un coût. Il s'agit de minimiser un coût global lors de l'affectation. Lorsqu'une formulation équivalente du problème comporte un nombre N d'utilisateurs égal à celui des sous porteuses, le problème revient à déterminer un couplage parfait et de poids minimal dans un graphe biparti ([Gondran&_79]). Un graphe biparti est un graphe dont l'ensemble de sommets peut être divisé en 2 ensembles distincts A et B tels que les arêtes du graphe relient un noeuds de A à un noeud de B . Un couplage du graphe est un sous ensemble d'arêtes qui n'ont aucun sommet en commun. Le couplage est dit parfait lorsque tous les sommets du graphe initial sont impliqués dans le couplage, il n'est possible que lorsque $|A|=|B|$. Autrement dit chaque élément de A est relié à un élément de B de manière unique. Le poids du couplage est égal à la somme du poids des arêtes.

Dans une autre formulation du problème, on recherche une permutation σ qui minimise $\sum c_{i\sigma(i)}$ parmi les $N!$ permutations possibles.

La solution optimale à ce problème est fournie par l'algorithme Hongrois crée par H. W. Kuhn en 1955 en hommage aux mathématiciens J. Egervary et D.König. C'est l'algorithme le plus efficace ($O(N^3)$) et de ce fait l'algorithme de référence pour la résolution optimale du problème formulé ([PapaSteig_82]).

Le principe de l'algorithme est le suivant: la (les) solution(s) optimale(s) sont inchangées en cas d'augmentation ou de diminution d'un même facteur sur tous les éléments d'une même ligne ou d'une même colonne de la matrice de coût. L'objectif de ces opérations est l'apparition de N zéros: un seul zéro par ligne et par colonne qui indiquent l'affectation optimale. [Issa&_03] fournit une illustration de l'application de l'algorithme.

Déroulement de l'algorithme:

Soit la matrice de coûts. 1) On soustrait le coût minimal de chaque ligne à tous les autres éléments de la ligne correspondante. 2) Dans les colonnes où il n'y a pas de zéros, On soustrait le coût minimal de la colonne à tous les autres éléments de la colonne. 3) On procède au marquage des zéros i.e. on tire d'un trait sur les lignes et les colonnes où il y a des zéros. Si le nombre minimal de traits nécessaires pour que chaque zéros soit marqué est égal à la dimension de la matrice on s'arrête. Sinon on va à l'étape 4. 4) Soustraire le coût le plus faible éléments non marqués à tous les éléments non marqués et le rajouter aux éléments marqués 2 fois (i.e. intersections d'une ligne rayés d'un trait et d'une colonne rayées d'un trait) puis on retourne à l'étape 3. A l'arrêt, les zéros indiquent l'affectation optimale.

Problème initialMatrice des coûts

V.P.	USINE			
	1	2	3	4
F	24	10	21	11
M	14	22	10	15
O	15	17	20	19
P	11	19	14	13

Ex. Le coût
d'affecter le
V.P. «P» à
l'usine 4 est
de 13.

Etape 1

V.P.	USINE				RÉDUIT DE:
	1	2	3	4	
F	14	0	11	1	10
M	4	12	0	5	10
O	0	2	5	4	15
P	0	8	3	2	11

Etape 2

V.P.	USINE				RÉDUIT DE:
	1	2	3	4	
F	14	0	11	0	
M	4	12	0	4	
O	0	2	5	3	
P	0	8	3	1	
	0	0	0	1	

Etape 3

V.P.	USINE			
	1	2	3	4
F	14	0	11	0
M	4	12	0	4
O	0	2	5	3
P	0	8	3	1

Donc, on
va à
l'étape 4.

Etape 4

V.P.	USINE			
	1	2	3	4
F	14	0	11	0
M	4	12	0	4
O	0	2	5	3
P	0	8	3	1

Valeur
minimum

Suite de l'Etape 4

V.P.	USINE			
	1	2	3	4
F	15	0	12	0
M	4	11	0	3
O	0	1	5	2
P	0	7	3	0

Etape 3

V.P.	USINE			
	1	2	3	4
F	15	0	12	0
M	4	11	0	3
O	0	1	5	2
P	0	7	3	0

Solution

V.P.	USINE			
	1	2	3	4
F	15	×	12	0
M	4	11	×	3
O	×	1	5	2
P	0	7	3	×

Annexe C : Le *waterfilling*

Expression du niveau d'eau et du vecteur de puissance

Considérons un unique utilisateur qui s'est vu allouer un ensemble de sous porteuses. On cherche à résoudre le problème suivant :

$$\begin{aligned} \max_{p_n} \sum_{n=1}^N \log_2(1+p_n\varphi_n) \\ \sum_{n=1}^N p_n = P_{T,\max} \\ h_n(p) \leq 0 \quad 1 \leq n \leq N \text{ avec } h_n(p) = -p_n \end{aligned}$$

Maximiser $\log_2(1+p_n\varphi_n)$ revient à minimiser l'opposé, on exprime le lagrangien qui s'exprime comme la somme de la fonction à minimiser et des fonctions contraintes associées à leur multiplicateur de Lagrange respectifs :

$$\mathcal{L}(p^1, \lambda, \mu) = - \sum_{n=1}^N \log_2(1 + p_n \varphi_n) + \lambda \left(\sum_{n=1}^N p_n - P_{T,\max} \right) + \sum_{n=1}^N \mu_n (-p_n)$$

Les conditions (nécessaires) de Kuhn et Tucker sont les suivantes ([p71, Bertsekas_82]) :

$$1 \leq n \leq N, \quad \frac{\partial \mathcal{L}(p^*, \lambda^*, \mu^*)}{\partial p_n} = 0 \quad (1)$$

$$1 \leq n \leq N, \quad \mu_n^* \geq 0 \text{ et } \mu_n^* h_n(p^*) = 0 \quad (2)$$

L'expression des dérivées partielles du lagrangien suivant la variable p est :

$$\frac{\partial \mathcal{L}(p^*, \lambda^*, \mu^*)}{\partial p_n} = \frac{\varphi_n}{(\ln 2)(1 + p_n \varphi_n)} + (\lambda^* - \mu_n^*)$$

La condition (1) donne l'expression de la puissance p_n du canal n :

$$p_n^* = \frac{1}{(\ln 2)(\lambda^* - \mu_n^*)} - \frac{1}{\varphi_n}$$

La condition (2) nous dit que si $p_n^* > 0$ alors $\mu_n^* = 0$. Cela donne l'intuition de la solution ci-dessous :

$$p_n^* = \left(\frac{1}{(\ln 2)\lambda^*} - \frac{1}{\varphi_n} \right)^+ \text{ où } x^+ \text{ désigne } x \text{ si } x > 0 \text{ et } 0 \text{ si } x < 0$$

$$\mu_n^* = \lambda^* - \frac{\varphi_n}{\ln 2} \text{ si } \left(\frac{1}{(\ln 2)\lambda^*} - \frac{1}{\varphi_n} \right) < 0 \text{ et } 0 \text{ sinon.}$$

Cette solution vérifie les conditions de Kuhn et Tucker. Ces conditions sont suffisantes car la fonction à minimiser est convexe et les contraintes sont convexes (car elles sont linéaires). Les vecteurs p et μ exprimés correspondent donc aux vecteurs optimaux p^* et μ^* .

Adoptons les notations qui suivent :

$\frac{1}{(\ln 2)\lambda^*} = \nu$; on a alors $p_n^* = \left(\nu - \frac{1}{\varphi_n} \right)^+$ et ν est le « waterlevel ». Enfin, $\mathcal{Q}$ est l'ensemble des canaux tels que $p_n^* \neq 0$ et on note $|\mathcal{Q}| = N^*$.

¹ p et μ sont des vecteurs de N éléments, ils ne seront pas soulignés pour alléger les notations ; λ est un scalaire.

Il reste à expliciter ν pour que l'expression du vecteur p^* (et μ^*) soit complète. Il suffit d'utiliser la première contrainte, on a alors (en insérant l'expression de p_n^* puis en développant) :

$$\sum_{n=1}^N p_n^* - P_{T,\max} = \sum_{n=1}^N \left(\nu - \frac{1}{\varphi_n} \right)^+ - P_{T,\max} = N^* \nu - \sum_{n \in \Omega} \frac{1}{\varphi_n} - P_{T,\max} = 0$$

$$\text{Finalement : } \nu = \frac{P_{T,\max} + \sum_{n \in \Omega} \frac{1}{\varphi_n}}{N^*}$$

Un algorithme de programmation du waterfilling

Principe

La tâche principale est la détermination du niveau d'eau noté ν . Plusieurs algorithmes sont proposés dans littérature, un algorithme itératif particulièrement simple est proposé par [PerezFon_05]. Pour déterminer le niveau d'eau, il faut trouver le zéro de la fonction

$$g(\nu) = \sum_{n=1}^N p_n(\nu) - P_{T,\max}. \text{ Etant donné l'expression de } p_n(\nu), g(\nu) = N\nu - \sum_{n=1}^N \frac{1}{\varphi_n} - P_{T,\max}. \text{ Tant que}$$

$g(\nu) > 0$, la contrainte en puissance n'est pas respectée. Cela signifie que certains canaux sont tellement mauvais que le budget de puissance alloué n'est pas suffisant. C'est pourquoi, ils doivent être abandonnés. Ainsi, à chaque étape lorsque $g(\nu) > 0$, on abandonne le canal qui présente le plus mauvais CgNR. Dès que $g(\nu) < 0$ on a atteint le nombre N^* des sous canaux (et surtout l'identification de ces sous canaux) pour lesquels la contrainte de puissance est respectée.

Déroulement

L'algorithme commence par classer les canaux en fonction de leur CgNR (noté λ sur Fig. 8). Appelons N_{sc} le nombre de sous canaux à chaque étape de l'algorithme. N_{sc} est initialisé à N .

L'estimation de ν à chaque étape est : $\nu_{\text{est}} = 1/\varphi_{N_{sc}}$. On calcule $g(\nu_{\text{est}}) = N_{sc}\nu_{\text{est}} - \sum_{n=1}^{N_{sc}} \frac{1}{\varphi_n} - P_{T,\max}$;

on a donc $g(\nu_{\text{est}}) = N_{sc}/\varphi_{N_{sc}} - \sum_{n=1}^{N_{sc}} \frac{1}{\varphi_n} - P_{T,\max}$. Si $g(\nu_{\text{est}})$ est positif, alors la puissance qui est

allouée à ce stade dépasse la contrainte, un sous canal est abandonné et $N_{sc} = N_{sc} - 1$. Lorsque $g(\nu_{\text{est}})$ est négatif, l'allocation est réalisable. On pose $N^* = N_{sc}$ et par conséquent ν et p_n^* sont complètement déterminés. Dans la figure 8, les canaux 6 et 7 ont été abandonnés.

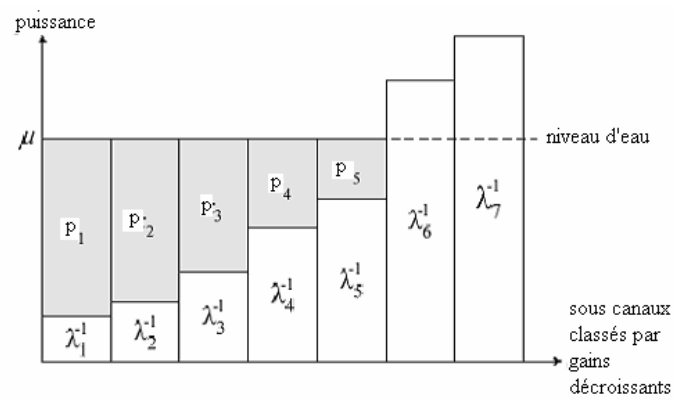


Figure 14: Niveau d'eau dans le *waterfilling*

La détermination du niveau peut être fastidieuse surtout si elle répétée plusieurs fois (c'est le cas en présence de plusieurs utilisateurs).

Références

- [Acena_05] Miguel Acena, Stephan Pfletsinger. "A Spectrally Efficient Method for Subcarrier and Bit Allocation in OFDMA". IEEE 61st Vehicular Technology Conference, 2005. VTC-Spring. 30 May-1 June 2005 Volume 3, Page(s): 1773 - 1777 Vol. 3.
- [Anas_04] Mohammad Anas. PHD Thesis, "QoS aware Subcarrier and Power Allocation in OFDMA Systems for Broadband Wireless Applications". Advisor:Kiseon Kim. 2004. Available at http://in.geocities.com/mohammad_anas/ms_thesis.pdf
- [benedetto_99] Sergio Benedetto, Ezio Biglieri Principles of Digital transmission with wireless Applications
- [Bertsekas_82] Dimitri P. Bertsekas. Computer science and applied mathematics: "Constrained optimization and Lagrange multiplier methods". Academic Press, Inc. 1982.
- [Ciblat_04] Philippe Ciblat, « Introduction à l'OFDM » ENST.
- [ChangKuo_04] Min-Kuan Chang, Jay Kuo. Power Control, Adaptive modulation and Subchannel Allocation for Multiuser downlink OFDM, 2004. http://viola.usc.edu/newextra/Publication/PDF/VTC/2004F_VTC_Chang.pdf
- [Chee&_04] Tommy K. Chee, Cheng-Chew Lim, Jinho Choi. "Sub-optimal Power Allocation for downlink OFDMA Systems". IEEE 60th VTC2004, 26-29 Sept. 2004, Page(s): 2015- 2019 Vol. 3. <http://ieeexplore.ieee.org/iel5/9623/30412/01400392.pdf?arnumber=1400392>
- [Chen&_04] Yung-Fang Chen, Jean-Wei Chen, Chih-Peng Li. "A Fast Suboptimal Subcarrier, Bit, and Power Allocation Algorithm for Multiuser OFDM-based Systems", *IEEE International Conference on Communications*, 20-24 June 2004, Volume: 6, pp. 3212- 3216 . <http://ieeexplore.ieee.org/iel5/9179/29123/01313137.pdf?arnumber=1313137>
- [Cho&_05] Myeon-gyun Cho, Woohyun Seo, Youngsoo Kim, Daesik Hong. "A Joint Feedback Reduction Scheme Using Delta Modulation for Dynamic Channel Allocation in OFDMA Systems". PIMRC 2005.
- [CoverThomas_91] Thomas M. Cover and Joy A. Thomas. "Elements of Information Theory". John Wiley and sons, Inc.1991.
- [Czylwik_96] Andreas Czylwik. "Adaptive OFDM for wideband radio channels". Global Telecommunications Conference, 1996. GLOBECOM '96. 'Communications: The Key to Global Prosperity Volume 1, 18-22 Nov. 1996. Page(s):713 - 718 vol.1. <http://ieeexplore.ieee.org/iel3/4477/12830/00594454.pdf?arnumber=594454>
- [DraHen_04] C. DRAGHICI et J.C. HENNETI. « UNE APPROCHE PAR COMPOSITION DU PROBLEME D'AFFECTATION DE ROTATIONS ». 5^e Conférence Francophone de Modélisation et Simulation. MOSIM'04 – du 1^{er} au 3 septembre 2004 - Nantes (France).
- [Ergen&_03] Mustafa Ergen, Sinem Coleri, Pravin Varaiya. "QoS Aware Adaptive Resource Allocation Techniques for fair Scheduling in OFDMA Based Broadband Wireless Access Systems", *IEEE Transactions on Broadcasting*, Vol.:49: Dec. 2003. http://paleale.eecs.berkeley.edu/~varaiya/papers_ps.dir/QoSAwareAdaptive.pdf
- [Gondran&_79] Michel Gondran, Michel Minoux. « Graphes et algorithmes ». 1979 Eyrolles.
- [Gross&_03] James Gross, Holger Karl, Frank Fitzek, Adam Wolisz. "Comparison of heuristic and optimal subcarrier assignment algorithms", *Proceedings of IEEE International Conference on Wireless Networks (ICWN) 2003*, pp. 249-255. http://www.tkn.tu-berlin.de/publications/papers/icwn_03.pdf
- [HuiZhou_06] Juhong Hui, YongxingZhou. "Enhanced Rate Adaptive Resource Allocation Scheme in Downlink OFDMA System". Vehicular Technology Conference, VTC2006-Spring. 7-10 may 2006.
- [IEEE802.16] IEEE 802.16-2004. Part 16: Air Interface for Fixed Broadband Wireless Access Systems—
- [Inyoung&_01] Inhyoung Kim, Hae Leem Lee, Beomsup Kim, Yong H. Lee. "On the use of linear programming for dynamic subchannel and bit allocation in multiuser OFDM", *IEEE Global Telecommunications Conference*, no. 1, Nov 2001, pp. 3648 – 3652. <http://ieeexplore.ieee.org/iel5/7633/20837/00966362.pdf>
- [Issa&_03] Issariyakul, T. Hossain, E. "Optimal radio channel allocation for fair queuing in wireless data networks". ICC '03. IEEE International Conference on Communications: 11-15 May 2003. Volume: 1, On page(s): 142- 146 vol.1.
- [JanLee_03] Jiho Jang, Kwang Bok Lee. "Transmit Power Adaptation for Multiuser OFDM Systems", *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 171-178, Feb. 2003. http://mobile.snu.ac.kr/mcl_list/papers/journal/jsac200302_jhjang_kblee.pdf
- [KimHan&_04] Keunyoung Kim, Hoon Kim, Youngnam Han. "Subcarrier and Power Allocation in OFDMA Systems". IEEE 60th Vehicular Technology Conference, VTC-Fall. 26-29 Sept. 2004, Volume 2, Page(s):1058 - 1062 Vol.2. <http://ieeexplore.ieee.org/iel5/9623/30411/01400183.pdf?arnumber=1400183>
- [KimKwak_04] Ho Seok Kim, Jin Sam Kwak, Jung Min Choi, and Jae Hong Lee. "Efficient Subcarrier and Bit Allocation Algorithm for OFDMA System with Adaptive Modulation", in *Proc. IEEE VTC Spring*, May 2004. <http://ieeexplore.ieee.org/iel5/9530/30274/01390570.pdf?arnumber=1390570>

[Kivanc&_00] Didem Kivanc, Hui Liu. "Subcarrier Allocation and Power Control for OFDMA", *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, 2000, pp. 147-151 vol.1.
<http://danube.ee.washington.edu/downloadable/didem/00910933.pdf>

[Kivanc&_03] Didem Kivanc, Guoqing Li, Hui Liu. "Computationally Efficient Bandwidth Allocation and Power Control for OFDMA", *IEEE Transactions on Wireless Communications*, , Nov 2003, vol. 2, no. 6, pp. 1150 – 1158.
<http://danube.ee.washington.edu/downloadable/didem/01244793.pdf>

[Koutsopoulos_02] Iordanis Koutsopoulos Advisor: Leandros Tassiulas. PHD Thesis: "Resource Allocation issues in broadband wireless networks with OFDM signalling", 2002. University of Maryland.

[lamy_00] Catherine Lamy, these ENST. « Communications à grande efficacité spectrale sur le canal à évanouissements », 2000
http://www.comelec.enst.fr/publications/publi/these_lamy.pdf

[Lawrey_99] E. Lawrey "Multiuser OFDM". International Symposium on Signal Processing and Its Applications, ISSPA 22-25 Aug '99. Proceedings of the Fifth Volume 2, Page(s):761 - 764 vol.2.
<http://ieeexplore.ieee.org/iel5/6605/17633/00815783.pdf?arnumber=815783>

[Lengoumbi&_06] Carle Lengoumbi, Philippe Godlewski, Philippe Martins. « An efficient Subcarrier Assignment Algorithm for Downlink OFDMA » VTC Fall 06, Montreal, Canada, 25-28 september 2006.

[Nakad_03] Jawad Nakad. "Allocations de ressources radio dans un réseau local sans fil (WLAN) de type OFDM. Mémoire de DEA. 2003. Faculté de Genie du Liban. <http://www.lb.refer.org/memoires/337626JawadNAKAD2.pdf>

[PapaSteig_82] Christos H Papadimitiou, Kenneth Steigltz. « Combinatorial Optimization, algorithms and complexity ». 1982 Prentice Hall.

[Peng&_05] Peng, Y.; Doufexi, A.; Armour, S.; McGeehan, J., "An investigation of dynamic sub-carrier allocation in OFDMA systems" Vehicular Technology Conference, 2005. VTC 2005-Spring. IEEE 61st Volume 3, 30 May-1 June 2005 Page(s):1808 - 1811 Vol. 3.

[PerezFon05] Daniel Perez Palomar, Javier R. Fonollosa. "Practical Algorithms for a Family of Waterfilling Solutions", *IEEE Trans. on Signal Processing*, Vol. 53, No. 2, pp. 686-695, Feb. 2005. <http://gps-tsc.upc.es/comm2/publications/PalomarTSP2005.pdf>

[Pfletschinger&_02] Stephan Pfletschinger, Joachim Speidel, Gerhard Münz. "Efficient Subcarrier Allocation for Multiple Access in OFDM Systems", 7th International OFDM-Workshop 2002 (InOWo'02), Hamburg, Germany, September 2002, pp. 21-25, 10-11.

[PietrykJan_02] Slawomir Pietrzyk, Gerard J.M. Janssen. "Multiuser Subcarrier Allocation for QoS provision in the OFDMA Systems", *IEEE 56th Vehicular Technology Conference Proceedings. VTC 2002-Fall*. pp. i- xl.
<http://www.ubicom.tudelft.nl/docs/PID11821.pdf>

[proakis_02] John G. Proakis, Masoud Salehi . Communication systems engineering 2nd edition, 2002

[RadioCom_04] Radio Communications – III: Autumn 2004. Multi-Carrier Based Multiple Access Schemes.
http://kom.aau.dk/~imr/RadioCommIII/2005/lect-4_autumn2005.pdf

[RheeCioffi_00] Wonjong Rhee, John M. Cioffi. "Increase in Capacity of Multiuser OFDM System Using Dynamic Subchannel Allocation", *Vehicular Technology Conference Proceedings. VTC 2000-Spring Tokyo*. pp.1085-1089 vol.2.
<http://ieeexplore.ieee.org/iel5/6849/18408/00851292.pdf>

[ShinsukePrasad_03] Hara Shinsuke, Ramjee Prasad. "Multimedia Techniques for 4G mobile communications". Artech House, 2003

[Wong&_04] Ian C. Wong, Zukang Shen, Jeffrey G. Andrews, Brian L. Evans. "A Low Complexity Algorithm for proportional Resource Allocation in OFDMA Systems", 2004. http://www.ece.utexas.edu/~jandrews/publications/WonShe_SIPS04.pdf

[WongMurch_99] Cheong Yui Wong, Roger S. Chen, Khaled Ben Letaief, Ross D. Murch. "Multiuser OFDM with Adaptive Subcarrier, Bit, and Power Allocation", *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 10, October 1999, pp. 1747 – 1758.
http://wsl.stanford.edu/~ee360/mu_ofdm.pdf

[WongTsui_99] Cheong Yui Wong, C. Y. Tsui, Roger S. Chen, Khaled Ben Letaief. "A Real-time Sub-carrier Allocation Scheme for Multiple Access Downlink OFDM Transmission", *Vehicular Technology Conference, 1999. Fall. IEEE VTS 50th* pp. 1124-1128 vol.2.

[Wyglinski_04] Alexander Wyglinski «Physical Layer Loading Algorithms for Indoor Wireless Multicarrier Systems». 2004, McGill university.

[YinLiu_00] Hujun Yin, Hui Liu. "An Efficient multiuser Loading Algorithm for OFDM-based Broadband Wireless Systems", *GLOBECOM 2000 - IEEE Global Telecommunications Conference*, no. 1, Nov 2000, pp. 103 – 107.
<http://ieeexplore.ieee.org/iel5/7153/19291/00891705.pdf?arnumber=891705>

[Yu&_06] Gunading Yu, Zhaoyang Zhang, Yan Chen, Ying Shi and Peiliang Qiu. "A novel Resource Allocation Algorithm for Real-time Services in Multiuser OFDM Systems". Vehicular Technology Conference, VTC2006-Spring. 7-10 may 2006.

[Zhang_04] G Zhang, "Subcarrier and bit allocation for real-time services in multiuser OFDM systems" IEEE ICC 04 Paris France, June 20-24, 2004, pp 2985_2989.

[Zhen&_03] Li Zhen Zhu Geqing Wang Weihua Song Junde. "Improved algorithm of multiuser dynamic subcarrier allocation in OFDM system". International Conference on Communication Technology Proceedings, ICCT: 9-11 April 2003, page(s): 1144- 1147 vol.2.
<http://ieeexplore.ieee.org/iel5/8586/27227/01209732.pdf?isnumber=&arnumber=1209732>