



**HAL**  
open science

# Allocation de Ressources dans les Réseaux Cellulaires pré-5G

Alexandre Ragaleux, Sébastien Baey

► **To cite this version:**

Alexandre Ragaleux, Sébastien Baey. Allocation de Ressources dans les Réseaux Cellulaires pré-5G. ALGOTEL 2016 - 18èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, May 2016, Bayonne, France. hal-01302444

**HAL Id: hal-01302444**

**<https://hal.science/hal-01302444v1>**

Submitted on 15 Apr 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Allocation de Ressources dans les Réseaux Cellulaires pré-5G

Alexandre Ragaleux<sup>1</sup> et Sébastien Baey<sup>1</sup>

<sup>1</sup>Sorbonne Universités, UPMC Univ Paris 06, CNRS, UMR 7606, LIP6, F-75005, Paris, France.

---

Cet article traite de l'allocation de ressources dans le sens descendant des réseaux pré-5G (LTE-B) en considérant les contraintes de la norme. Le problème consiste, à un instant donné, à affecter aux flux utilisateurs les ressources radios (divisées dans le domaine fréquentiel) de manière à maximiser une fonction d'utilité.

Ce problème étant NP-Difficile, nous proposons un algorithme d'approximation. Cet algorithme est composé de trois routines qui fonctionnent ensemble afin d'allouer efficacement les ressources. Les résultats, obtenus par simulation, montrent que notre solution permet d'obtenir une très bonne équité entre les utilisateurs tout en maintenant une capacité proche de la solution optimale.

**Mots-clés :** allocation de ressources, LTE-B, algorithme d'approximation.

---

## 1 Introduction

**Contexte.** Dans cet article, nous nous intéressons à la partie *réseaux d'accès radio* de LTE-B qui a été publiée au début de l'année 2016. Nous considérons une station de base qui communique avec un certain nombre d'utilisateurs mobiles. Il peut s'agir indifféremment d'une *macro-cell* ou d'une *small-cell*. Dans le sens descendant de LTE-B, les ressources radios qui sont utilisées par la station de base pour transmettre des données aux utilisateurs sont des blocs de sous-porteuses. Ainsi, le problème consiste à partager ces ressources entre les différents flux utilisateurs de façon à maximiser un objectif donné. De plus, ce processus d'allocation de ressources est périodique de période égale à la durée d'un TTI (*Transmission Time Interval*), de valeur 1 ms [3GP16].

Une particularité des réseaux sans fil est d'être soumis à de nombreuses déficiences du canal (atténuation du signal, évanouissement sélectif en fréquence, multi-trajets ...). Par conséquent, à un même instant, les ressources peuvent être de qualités variables du point de vue d'un utilisateur. Sur la base des rapports signaux à bruits (*Signal to Noise Ratio* (SNR)) perçus par chaque utilisateur sur chacune des ressources, la station de base adapte dynamiquement sa modulation et son codage (*Modulation and Coding Scheme* (MCS)) afin de transmettre la plus grande quantité de données possible sans dépasser un *taux d'erreur binaire* seuil. Dans la suite de l'article, nous considérerons l'hypothèse des *utilisateurs gourmands*, i.e. nous supposons que la station de base a toujours des données à transmettre aux utilisateurs.

**Notations.** Nous appelons SU (*Scheduling Unit*) la plus petite unité de ressources qui peut être allouée à un utilisateur. L'ensemble de tous les SUs est noté  $\mathcal{K} = \{k_1, k_2, \dots, k_K\}$ , où  $K$  est le nombre de SUs. De même, nous notons  $\mathcal{U} = \{u_1, u_2, \dots, u_U\}$  l'ensemble des utilisateurs avec  $U$  le nombre d'utilisateurs actifs dans la cellule. Ainsi, à chaque TTI, l'ordonnanceur va allouer à chaque utilisateur  $u$  un sous-ensemble de  $\mathcal{K}$  noté  $N_u$  (sous la contrainte que tous les ensembles  $N_u$  soient disjoints). Enfin, nous notons  $\mathcal{M} = \{m_1, m_2, \dots, m_M\}$  l'ensemble des MCS utilisables, et  $M$  le nombre total de MCS. Le MCS le plus efficace qui peut être utilisé par la station de base sur le  $SU_k$  pour l'utilisateur  $u$  (sans enfreindre la contrainte sur le taux d'erreur binaire) est noté  $m_{u,k}$ .

**Instance du problème.** Les données du problème peuvent être décrites par une matrice  $I$  représentant les

valeurs des MCS  $m_{u,k}$  pour chaque couple (*utilisateur, SU*) :

$$I = (m_{u,k})_{u \in \mathcal{U}, k \in \mathcal{K}} = \begin{pmatrix} m_{u_1, k_1} & \cdots & m_{u_1, k_K} \\ \vdots & \ddots & \vdots \\ m_{u_U, k_1} & \cdots & m_{u_U, k_K} \end{pmatrix}.$$

**La contrainte du *Single MCS*.** Pour chaque utilisateur, la station de base ne peut utiliser qu'un seul MCS sur tous les SUs qui sont alloués à cet utilisateur pour un TTI [3GP16]. Ceci est à mettre en opposition avec un système OFDMA classique, dans lequel la station de base peut transmettre avec des MCS différents sur chacune des ressources de façon à maximiser le débit obtenu par l'utilisateur. Cette contrainte réduit donc considérablement la liberté dans la manière d'allouer les ressources, et la diversité fréquentielle ne peut être totalement exploitée. À notre connaissance, la contrainte du *Single MCS* a été très peu considérée par la communauté [LPC<sup>+</sup>12, LCC14]. Dans la suite, nous considérons que chaque utilisateur est contraint d'utiliser le MCS associé au SU de plus mauvaise qualité qui lui est alloué, ceci afin de respecter le *taux d'erreur* visé sur chacune de ses ressources. Nous désignons par  $m_u$  le MCS utilisé pour transmettre sur les ressources allouées à l'utilisateur  $u$ . Ainsi, la contrainte du Single MCS s'exprime par  $m_u = \min(\{m_{u,k} \mid k \in N_u\}), \forall u$ .

**Calcul du débit.** Avec LTE-B, le débit utile est donné par un tableau indexé par deux paramètres [3GP16] :

- l'ensemble des SUs alloués à l'utilisateur ( $N_u$  pour l'utilisateur  $u$ ),
- la qualité globale des ressources allouées à cet utilisateur (MCS  $m_u$  pour l'utilisateur  $u$ ).

Soit  $D_m(N)$  le débit utile (et instantané) obtenu par un utilisateur en lui allouant le sous-ensemble  $N \subset \mathcal{K}$  et le MCS  $m$ . Pour un  $m$  fixé, notons que  $D_m(N)$  possède les propriétés d'une fonction super-additive, i.e. pour tous sous-ensembles disjoints  $N_1, N_2 \subset \mathcal{K}$ , alors  $D_m(N_1 \cup N_2) \geq D_m(N_1) + D_m(N_2)$  [3GP16]. Cette propriété spécifique à LTE rend le problème plus difficile car non-linéaire.

**Objectif de l'allocation.** Afin de rester général, nous introduisons la notion de fonction d'utilité  $\varphi_u$  de l'utilisateur  $u$ . L'expression exacte de  $\varphi_u$  dépend de l'objectif poursuivi par l'opérateur. Cependant, nous considérons que chaque fonction  $\varphi_u$  prend, au moins,  $N_u$  et  $m_u$  comme paramètres. L'objectif est donc de trouver des sous-ensembles disjoints  $N_{u_1}, N_{u_2}, \dots, N_{u_U}$  tels que  $\sum_{u \in \mathcal{U}} \varphi_u(N_u, m_u)$  soit maximisé sous la contrainte du Single MCS. De plus, pour un  $m_u$  fixé, nous supposons que chaque fonction  $\varphi_u$  est une fonction croissante du cardinal de son premier paramètre. Dans la suite, nous désignons par **P1** ce problème.

**Exemple d'allocation.** Supposons que l'on souhaite maximiser la capacité du système. Pour cela, nous définissons la fonction d'utilité par :  $\varphi_u(N_u, m_u) = D_{m_u}(N_u)$ . Considérons un système avec 3 SUs  $\mathcal{K} = \{k_1, k_2, k_3\}$  et deux utilisateurs  $\mathcal{U} = \{u_1, u_2\}$ . De plus, supposons que l'ordonnanceur alloue les deux premières ressources à  $u_1$  et la dernière ressource à  $u_2$  (i.e.  $N_{u_1} = \{k_1, k_2\}$  et  $N_{u_2} = \{k_3\}$ ). Par conséquent, les *Single MCS* de chaque utilisateur sont choisis de cette façon :  $m_{u_1} = \min(m_{u_1, k_1}, m_{u_1, k_2})$  et  $m_{u_2} = m_{u_2, k_3}$ . Le débit total du système de cette solution est donc égal à  $D_{m_{u_1}}(N_{u_1}) + D_{m_{u_2}}(N_{u_2})$ .

**Motivation et contribution.** Dans cet article, nous proposons un algorithme proche du standard LTE-B qui peut donc être implémenté facilement dans un système réel. Ainsi, nous considérons les spécificités de LTE rarement prises en compte dans la littérature : (1) la contrainte du *Single MCS* et (2) le calcul non-linéaire des débits. Même si une grande avancée a été réalisée dans [LCC14] avec la considération de la contrainte du Single MCS dans le cadre des réseaux LTE-A, à notre connaissance, aucun algorithme ne prend en compte ces deux particularités comme partie intégrante d'une procédure d'allocation de ressources. De plus, notre solution est générique, et permet donc d'adresser une grande variété d'objectifs.

## 2 Algorithme

**Première étape (principe).** Le problème, tel que formulé, est difficile à résoudre. Dans un premier temps, nous travaillons donc sur une version relaxée de **P1**. Ainsi, nous définissons **P2** de la même manière que **P1** sans la contrainte du Single MCS. Les utilisateurs peuvent donc utiliser autant de MCS qu'ils le souhaitent. Si nous reprenons l'exemple précédent, l'utilisateur  $u_1$  peut maintenant recevoir des données sur la ressource  $k_1$  avec le MCS  $m_{u_1, k_1}$  tout en recevant des données sur la ressource  $k_2$  avec le MCS  $m_{u_1, k_2}$ . Si  $m_{u_1, k_1} \neq m_{u_1, k_2}$ , le débit total obtenu est donc égal à  $D_{m_{u_1, k_1}}(\{k_1\}) + D_{m_{u_1, k_2}}(\{k_2\}) + D_{m_{u_2, k_3}}(\{k_3\})$ . On

peut alors penser que le problème devient “facile” mais la non-linéarité du calcul du débit rend le problème toujours difficile. À noter qu’une instance de **P1** est également une instance valide pour **P2**.

Pour résoudre **P2**, nous nous appuyons sur le **problème du social welfare** (SWP) [MSV08] car ce dernier est très proche de **P2**. Nous montrons que SWP et **P2** sont équivalents et donc, comme SWP est NP-complet, alors **P2** et **P1** le sont aussi.

Nous cherchons donc à approximer une solution optimale pour **P2**. Pour ce faire, notre première routine interprète l’instance de **P1** comme une instance de **P2**, et retourne donc une solution approximée pour **P2**. Plus précisément, cette routine réduit l’instance de **P2** en une instance de SWP équivalente. Une solution sous-optimale de l’instance SWP est ensuite calculée en utilisant un algorithme d’approximation de la littérature qui dépend de la classe des fonctions d’utilité. Pour ne pas se limiter à une classe de fonction précise, on note  $\alpha$  le facteur d’approximation de l’algorithme utilisé pour résoudre **P2**. Pour la classe des fonctions super-additives  $\alpha = \sqrt{\log(K)}/K$ . L’algorithme retourne  $\{N_{u,m}\}_{u,m}$ , avec  $N_{u,m}$  l’ensemble des ressources allouées à utilisateur  $u$  pour le MCS  $m$ .

**Deuxième étape.** Étant donné une solution déterminée par la première routine  $\{N_{u,m}\}_{u,m}$ , nous cherchons maintenant à prendre en compte la contrainte du *Single MCS*.

Pour chaque utilisateur  $u$ , cette deuxième routine consiste à sélectionner le “meilleur MCS” en considérant les sous-ensembles  $N_{u,m_1}, N_{u,m_2}, \dots, N_{u,m_M}$ . Concrètement, on détermine  $m_u$  le Single MCS de l’utilisateur  $u$  et  $N_u$  le sous-ensemble des ressources allouées à cet utilisateur comme suit :

$$m_u = \operatorname{argmax}_m \Phi_u \left( \bigcup_{m' \geq m} N_{u,m'}, m \right) \quad \text{et} \quad N_u = \bigcup_{m' \geq m_u} N_{u,m'}.$$

De cette façon,  $m_u$  respecte bien la contrainte du Single MCS par rapport à l’ensemble  $N_u$ . Le théorème suivant donne une garantie (en facteur constant) sur la qualité de l’allocation par rapport à la solution optimale.

**Théorème 1.** *La deuxième routine est une  $\frac{\alpha}{M}$ -approximation pour **P1**.*

*Idee de la preuve.* Considérons une instance à la fois valable pour **P1** et **P2**. Soit  $V_{\mathbf{P2}} = \sum_u \sum_m \Phi_u(N_{u,m}, m)$ , le coût de notre première routine pour **P2** (et  $V_{\mathbf{P2}}^*$  le coût optimal). De même, soit  $V_{\mathbf{P1}} = \sum_u \Phi_u(N_u, m_u)$  le coût de notre solution pour **P1** (et  $V_{\mathbf{P1}}^*$  le coût optimal). La preuve consiste à borner  $V_{\mathbf{P1}}^*$  par une expression de  $V_{\mathbf{P1}}$  jusqu’à obtenir  $V_{\mathbf{P1}}^* \leq M/\alpha \times V_{\mathbf{P1}}$  (en utilisant le fait que  $\sum_m \Phi_u(N_{u,m}, m) \leq \sum_m \Phi_u(\bigcup_{m' \geq m} N_{u,m'}, m)$ ,  $\forall u$ ).

**Troisième étape.** Après l’exécution de la deuxième étape, un MCS a été affecté à chaque utilisateur ( $m_u$  pour l’utilisateur  $u$ ). Les SUs alloués par la première routine qui ne supportent pas ce MCS ne seront pas utilisés et seront donc perdus. Par conséquent, l’algorithme précédent peut amener à un gaspillage de ressources.

L’objectif de cette troisième étape est donc de re-distribuer efficacement les ressources non-allouées. Pour cela, nous notons  $\mathcal{N}^r$  l’ensemble des SUs non-alloués après l’exécution des deux premières routines. Pour chaque couple  $(u, m)$ , nous construisons  $N_{u,m}^r$  qui est l’ensemble des ressources de  $\mathcal{N}^r$  tel que  $\forall k \in N_{u,m}^r$  nous avons  $m_{u,k} \geq m$ . Notons que dans ce cas, les ensembles  $\{N_{u,m}^r\}$  ne sont pas nécessairement disjoints. A chaque étape et pour chaque utilisateur  $u$ , nous lui associons une métrique  $g_u$  appelée *gain par ressource* et calculée de la façon suivante :

$$g_u = \max_{m \leq m_u} \frac{\Phi_u(N_{u,m}^r \cup N_u, m) - \Phi_u(N_u, m_u)}{|N_{u,m}^r|}, \quad \text{avec } |N_{u,m}^r| \text{ le cardinal de } N_{u,m}^r.$$

Soit  $m'_u$  le MCS qui maximise l’expression ci-dessus. L’utilisateur pour lequel la valeur de  $g_u$  est la plus grande est appelé l’utilisateur *élu*. On affecte alors à ce dernier le MCS  $m'_u$  et on lui alloue l’ensemble  $N_{u,m'_u}^r$  en plus de  $N_u$ . Les ensembles  $\mathcal{N}^r$  et  $N_{u,m}^r(\forall u, m)$  sont ensuite reconstruits et la procédure se répète. Cette troisième routine se termine lorsque plus aucune allocation n’est bénéfique, i.e.  $\forall u, g_u \leq 0$ . Notons que cette étape améliore la deuxième routine sans modifier le facteur d’approximation de l’algorithme complet.

**Un mot sur la complexité.** L’algorithme proposé est constitué de routines de complexité polynomiale. Il en résulte que notre algorithme est également de complexité polynomiale. Si les fonctions d’utilité sont super-additives, on peut montrer que la complexité est en  $O(UK \log(K) + U^2MK)$ , ce qui est adapté à LTE-B.

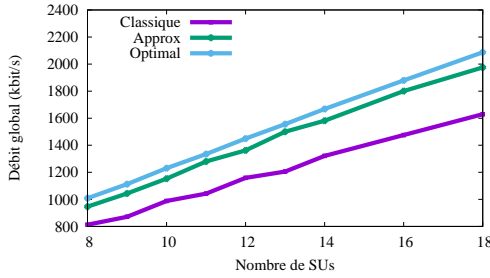


FIGURE 1: Débit global du système vs nombre de ressources.

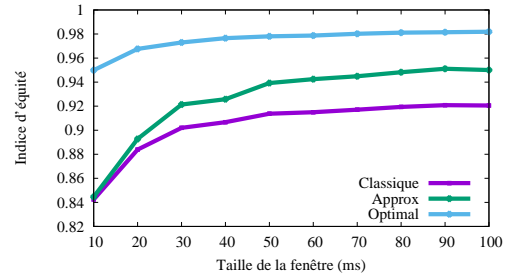


FIGURE 2: Indice d'équité vs taille de la fenêtre.

### 3 Évaluation de performance

Dans cette section, nous considérons uniquement la fonction d'utilité  $\varphi_u(N_u, m_u) = D_{m_u}(N_u)/\bar{D}_u$  qui exprime le débit instantané de l'utilisateur  $u$  divisé par son débit moyen  $\bar{D}_u$  (calculé sur une fenêtre glissante). Cette fonction d'utilité est dérivée du *Proportional Fair* (PF) [LPC<sup>+</sup>12] et permet d'assurer l'équité entre les utilisateurs. Les résultats ont été obtenus par simulation à l'aide du logiciel Opnet. Les principaux paramètres de simulation sont donnés dans le Tableau 1. Le nombre d'utilisateurs est volontairement limité car l'algorithme optimal est de complexité exponentielle ( $O(U^N)$ ).

Cependant, des résultats aux tendances similaires pour une instance de 50 utilisateurs ont également été observés. Pour évaluer l'équité entre les utilisateurs, nous utilisons l'*Indice de Jain* défini par :

$$F(\Delta t) = \frac{(\sum_{u \in \mathcal{U}} \bar{D}_u(\Delta t))^2}{U \cdot \sum_{u \in \mathcal{U}} (\bar{D}_u(\Delta t))^2},$$

avec  $\bar{D}_u(\Delta t)$  le débit moyen de l'utilisateur  $u$  calculé sur des fenêtres de durée  $\Delta t$ .

La Figure 1 compare notre algorithme d'approximation (APPROX) avec (1) l'algorithme optimal prenant en compte la contrainte du *Single MCS* et (2) l'algorithme classique, i.e. l'algorithme optimal qui ne considère pas les spécificités de LTE-B comme partie intégrante du processus d'allocation. Cependant, la contrainte du *Single MCS* est tout de même prise en compte après exécution de l'algorithme. La Figure 2 montre l'indice d'équité en fonction de  $\Delta t$  dans un système composé de 18 SUs. On peut remarquer que le débit global obtenu par notre algorithme reste très proche de la solution optimale, et se montre bien supérieur à l'algorithme classique. Bien sûr, la solution optimale obtient le meilleur débit et la meilleure équité. L'équité obtenue avec notre algorithme est tout de même très convenable (environ 0,94). L'algorithme proposé se révèle donc efficace en pratique.

**Conclusion.** Nous avons construit un algorithme d'approximation efficace et conforme à la norme pour l'allocation de ressources dans le sens descendant de LTE-B. Par ce biais, nous avons montré que la prise en compte des spécificités de LTE-B est essentielle afin d'obtenir des résultats performants.

### Références

- [3GP16] 3GPP. Physical layer procedures (Release 13). TS 36.213, 3GPP, 2016.
- [LCC14] Hong-Sheng Liao, Po-Yu Chen, and Wen-Tsuen Chen. An efficient downlink radio resource allocation with carrier aggregation in lte-advanced networks. *IEEE TMC*, 13, Oct 2014.
- [LPC<sup>+</sup>12] Suk-Bok Lee, I. Pefkianakis, S. Choudhury, Shugong Xu, and Songwu Lu. Exploiting spatial, frequency, and multiuser diversity in 3gpp lte cellular networks. *IEEE TMC*, Nov 2012.
- [MSV08] Vahab Mirrokni, Michael Schapira, and Jan Vondrak. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. *ACM Conf. on Electronic Commerce*, 2008.

Paramètre	Valeur
Modèle du canal	Atténuation avec évanouissement de <i>Rayleigh</i>
Diamètre de la cellule	1 km
MCS utilisés	QPSKr0.3, QPSKr0.58, 16QAMr0.36, 16QAMr0.48, 64QAMr0.45, 64QAMr0.65
Nombre d'utilisateurs	8 utilisateurs
Nombre de TTI	2000
Modèle de mobilité	<i>Random Way Point</i>

Tableau 1: Paramètres de simulation.